

UNIVERSIDADE DE LISBOA
Faculdade de Ciências
Departamento de Informática



**FERRAMENTA PARA ANÁLISE DE MODOS E
EXPERIÊNCIAS DE LEITURA**

Artemisa Maria Mendes Moreno

DISSERTAÇÃO

MESTRADO EM ENGENHARIA INFORMÁTICA
Especialização em Sistemas de Informação

2012

UNIVERSIDADE DE LISBOA
Faculdade de Ciências
Departamento de Informática



**FERRAMENTA PARA ANÁLISE DE MODOS E
EXPERIÊNCIAS DE LEITURA**

Artemisa Maria Mendes Moreno

DISSERTAÇÃO

Projecto orientado pelo Prof. Doutor Nuno Manuel de Carvalho Guimarães
e co-orientado pelo Prof. Doutor Carlos Alberto Pacheco dos Anjos Duarte

MESTRADO EM ENGENHARIA INFORMÁTICA
Especialização em Sistemas de Informação

2012

Agradecimentos

Em primeiro lugar, queria agradecer aos meus pais, Vítor e Elizabeth por todo o apoio que me deram, por terem acreditado em mim e pela paciência que tiveram comigo durante todos estes anos. Queria agradecer aos meus irmãos Smaíellô e Liniane, a minha avó, meus tios e primos que apesar de estarem longe sempre me apoiaram.

Agradeço também ao meu orientador, Professor Nuno Guimarães pela paciência que teve comigo e pela sua excelente orientação durante a realização do trabalho.

Agradeço também aos meus colegas de grupo Sheilla Simões e Badjinca Baticam que estiveram comigo durante todo o percurso académico. Aos meus amigos e colegas da faculdade, que me acompanharam durante todos os anos, com especial destaque para Eloisa Marques, Edna Baessa, David Marques, Maria Santos e Daniela Barbosa, um muito obrigado.

Um muito obrigado a todas as pessoas que sempre estiveram comigo no decorrer do meu percurso de vida académica e no desenvolvimento deste trabalho.

Aos meus pais e meus irmãos

Resumo

O estudo de processos de leitura em suporte digital tem adquirido importância na sequência da disseminação de novos tipos de dispositivos. Este projeto tem como objetivo a análise de modos e de experiências de leitura, ou seja o processo de avaliar a experiência pessoal do utilizador na atividade de leitura. Para isso é desenvolvido uma ferramenta que visa facilitar a análise de aspetos particulares da leitura através da conceção de experiências em que são modificados modos de apresentação dos textos. A ferramenta apresenta opções que permite a avaliação da experiência dos utilizadores. O processo da avaliação começa com a preparação de um conjunto de textos base que vão ser lidos por um conjunto de utilizadores. A preparação de textos consiste em determinar as formas de visualização de textos, que podem ser: leitura contínua, segmentação palavra-por-palavra(segmentação 1-1), segmentação palavra-por-palavra mostrando simultaneamente as palavras anterior e seguinte(segmentação 3-3) e segmentação manual. No final de cada leitura é apresentada um conjunto de questões que servem para avaliar a experiência dos utilizadores.

Palavras-chave: Experiência de leitura, leitura, segmentação, UX

Abstract

The study of processes in digital reading has gained importance with the spread of new types of devices.

This project aims to analyze modes and reading experiences, ie the process of evaluating the user experience in reading activity. For this, a tool was developed to facilitate the analysis of particular aspects of reading through the design of experiments that create modified forms of presentation of texts. The tool has options to evaluate the user experience. The evaluation process begins with the preparation of a set of basic texts that will be read by a set of users. The preparation of texts determines the forms of display of text, which may be: continuous reading, word-by-word segmentation (segmentation 1-1), word-by-word presented simultaneously with the previous and following words(segmentation 3-3) and segmentation manual. At the end of each reading a set of questions is presented and used to evaluate the user experience.

Keywords: Reading experience, reading, segmentation, UX

Conteúdo

Lista de Figuras	xv
Lista de Tabelas	xvii
Listagem	xix
1 Introdução	1
1.1 Motivação	1
1.2 Objetivos	2
1.3 Estrutura do documento	2
2 Trabalho relacionado	5
2.1 User-eXperience (UX)	5
2.2 Eye-traking e Leitura	7
2.3 Aplicações relacionadas	9
2.3.1 TextPlayer Lite (Autoscroll)	9
2.3.2 Reader Lite	10
2.3.3 Reading Player Lite	11
2.4 A análise de leitura	12
2.5 Resumo	14
3 Trabalho desenvolvido	15
3.1 Análise de Requisitos	15
3.1.1 Requisitos funcionais	15
3.1.2 Requisitos não funcionais	16
3.1.3 Casos de Uso	17
3.2 Segmentação	18
3.2.1 Segmentação 1-1	19
3.2.2 Segmentação 3-3	19
3.2.3 Segmentação manual	19
3.3 Implementação	24
3.3.1 Ambiente de programação	24

3.3.2	Interface para sistemas externos	25
3.3.3	TextReaderA	25
3.4	Concretização	30
3.5	Plano de desenvolvimento	34
4	Experiências e Resultados	35
4.1	Descrição das experiências	35
4.1.1	Questionário utilizado na avaliação	35
4.2	Análise de Resultados	37
4.2.1	Comparação dos resultados das experiências	42
5	Conclusão	45
	Bibliografia	49
	Abreviaturas	51
A	Mapa de Gantt	53
B	Poster: Análise de Modalidades e Modos de Leitura Contínua	55
C	Textos utilizados nas experiencias	57
D	Código fonte	59

Lista de Figuras

2.1	TextPlayer(Autoscroll)	10
2.2	Reader Lite	10
2.3	Reading Player Lite	12
3.1	Casos de uso	18
3.2	Imagem de texto com movimentos oculares	20
3.3	Combinação de imagens de movimento ocular de vários observadores	21
3.4	Ambiente de desenvolvimento da aplicação TextReaderA	24
3.5	TextReaderA (janela principal)	26
3.6	TextReaderA (Leitura contínua)	27
3.7	TextReaderA (Leitura com segmentação 1-1)	28
3.8	TextReaderA (Leitura com segmentação 3-3)	28
3.9	TextReaderA (Leitura com segmentação manual)	29
4.1	Tempo de leitura vs. Tipo de segmentação	42
4.2	Feedback dos utilizadores relativamente a segmentação	42
4.3	Feedback dos utilizadores relativamente a qualidade da aplicação	43

Lista de Tabelas

3.1	Tabela de segmentação	22
4.1	Resultados da Experiência 1 (Leitura Contínua)	38
4.2	Resultados da Experiência 2 (Segmentação 1-1)	39
4.3	Resultados da Experiência 3 (Segmentação 3-3)	40
4.4	Resultados da Experiência 4 (Segmentação manual)	41

Listings

3.1	Ficheiro XML com a segmentação manual	23
3.2	Ficheiro XML com o questionário utilizado na aplicação	25
D.1	MainClass	59
D.2	Codigo 2: PainelPrincipal.inicializar()	59
D.3	PainelPrincipal.setPainelTexto(String seg)	61
D.4	PainelPrincipal.createAndShowGUI()	62

Capítulo 1

Introdução

1.1 Motivação

A interação com sistemas computacionais inclui frequentemente processos de leitura, sejam eles de reconhecimento de palavras e frases ou de leitura atenta e contínua. A conceção da interface dos sistemas computacionais é determinante para satisfazer tanto as condições de usabilidade e acessibilidade como para enriquecer a experiência dos utilizadores (*user experience* a.k.a UX).

O conceito de UX tem sido amplamente divulgado e rapidamente aceite na comunidade de Interação Humano-Computador (HCI), no entanto sem que seja claramente definido [1].

A apresentação de texto digital é um problema antigo, emergente desde que se tornou possível e comum a apresentação de textos nos ecrãs com resoluções suficientes [2]. A leitura em geral consiste fundamentalmente em ver e compreender palavras ou frases, mas as pessoas podem ler por diferentes razões e de maneiras diferentes [2]. A leitura, ou o processo de leitura mais especificamente, tem sido objeto de análise e sistematização [3] e [4], embora estes estudos sejam pouco operacionais na conceção dos elementos de interação.

Este trabalho enquadra-se na análise dos modos e da atividade de leitura, e tem como contexto mais lato o projeto de investigação RAPS¹-*Reading Analysis with Physiological Signals*. Este projeto tem como objetivo estudar os processos neuro-fisiológicos subjacentes à leitura atenta e contínua, sendo particularmente relevante a distinção entre a leitura de palavras isoladas, de segmentos de texto ou de um texto contínuo. A compreensão do processo de leitura e a sua base neural tem sido amplamente abordado em vários estudos [5], [6] e [7]. O movimento dos olhos em particular tem sido analisado extensivamente usando sistemas de eye-tracking, para inferir a natureza dos processos cognitivos relevantes para a atividade de leitura. Diferentes tipos de movimentos dos olhos podem ser registados durante a leitura. Por outro lado, a leitura contínua e atenta é uma experiência

¹PTDC/EIA-EIA/113660/2009 - RAPS

humana que naturalmente funde processos em múltiplos níveis e ativa simultaneamente vários caminhos cerebrais. A ferramenta **TextReaderA** foi desenvolvida para explorar o suporte à análise da experiência de leitura. Esta ferramenta apresenta textos simples de acordo com diferentes estilos de segmentação. Além de permitir ao utilizador configurar o modo de leitura do texto de acordo com diferentes estilos de segmentação, esta ferramenta também permite a realização de outras operações, por exemplo, mudar a fonte e o tamanho do texto, alterar as cores do fundo e das letras e configurar a velocidade de apresentação de texto. A ferramenta também procura avaliar a experiência do utilizador (UX).

1.2 Objetivos

O objetivo principal deste trabalho é contribuir para compreender e avaliar a experiência de leitura (*reading experience*, a.k.a RX). Para isso foi criada uma ferramenta (TextReaderA) que permite variar as características de leitura e medir o efeito dessas variações. A ferramenta deve ser capaz de fornecer informações que permitem analisar vários modos de leitura e avaliar as experiências de leitura dos utilizadores.

Em primeiro lugar, a ferramenta TextReaderA tem que apresentar opções que permitam ao utilizador configurar o modo de leitura que pretender. O modo de leitura pode ser leitura contínua ou leitura segmentada, em que esta pode ser leitura por palavra isolada ou segmentos do texto. A ferramenta deve permitir ao utilizador escolher o texto que pretende ler, configurar a fonte e o tamanho das letras, alterar as cores do fundo e do texto e alterar a velocidade de apresentação de texto para o caso da leitura segmentada.

Em segundo lugar, a ferramenta deve disponibilizar informações que permitem avaliar a experiência de leitura. Para isso, no final de cada leitura de um texto é apresentado ao utilizador um questionário que lhe permite avaliar a sua experiência de leitura.

Em suma, o objetivo deste trabalho é criar uma ferramenta que permite analisar os diferentes modos de leitura, e compreender e avaliar as experiências de leitura dos utilizadores.

1.3 Estrutura do documento

Este documento está organizado da seguinte forma:

- Capítulo 2 – Trabalho Relacionado

Descreve trabalhos relacionados com a área em que o projeto se insere.

- Capítulo 3 – Trabalho desenvolvido

Descreve os requisitos funcionais do sistema assim como os caso de uso.

Este capítulo descreve todo o trabalho desenvolvido. Desde análise de requisitos, descrição do ambiente de programação, interfases para sistemas externos, os tipos de seguimentação e a implementação das funcionalidades.

- Capítulo 4 –Experiências e Resultados

Este capítulo descreve as experiências realizadas na avaliação de leitura assim como a análise de resultados obtidos.

- Capítulo 5 – Conclusão Neste capítulo apresentam-se as conclusões do trabalho.

- Bibliografia

- Apêndices

Capítulo 2

Trabalho relacionado

Este capítulo apresenta alguns conceitos e tecnologias relacionados com as diferentes áreas em que o projeto se insere. O trabalho desenvolvido está relacionado com a análise de modos e de experiências de leitura dos utilizadores. Para enquadrar o trabalho, é necessário compreender melhor o que significa UX, apresentando-se um estudo [1] cujo principal objetivo é compreender a definição de UX. Apresentamos ainda uma secção que descreve o uso da técnica de Eye-tracking na análise de leitura. Por fim, são apresentadas algumas aplicações relacionadas com o projeto desenvolvido, em particular algumas que são usadas em dispositivos móveis, TextPlayer (Autoscroll) ou Reader Lite.

2.1 User-eXperience (UX)

Nesta secção é apresentado o estudo referido anteriormente [1], que tem como objetivo ajudar a compreensão do conceito de UX. Este estudo revela que, apesar do crescente interesse na experiência do utilizador (UX), tem sido difícil obter um acordo comum sobre a sua natureza e o seu âmbito. A validade da UX é vista como algo desejável, mas o que exatamente significa é algo que permanece aberto e discutível. Nos últimos anos, tem sido realizadas conferências, workshops, fóruns e atividades similares com o objetivo de entender melhor o conceito da UX e desenvolver uma visão unificada sobre UX [8].

O estudo em questão tem como objetivo principal promover discussões sobre a natureza da UX. Para isso reuniu pontos de vista sobre UX de 275 pesquisadores e profissionais da academia e da indústria. O questionário utilizado na pesquisa foi dividido em três secções: Declarações, Definições e Backgrounds. Na secção "Declarações UX" foram apresentadas 23 afirmações e os entrevistados tinham que indicar o seu nível de concordância com essas afirmações. Na secção "Definições UX", os entrevistados tinham que expressar as suas opiniões sobre conjunto de cinco definições, o que gostavam ou não nessas definições e como achavam que podiam ser melhoradas. Na secção *Background* dos entrevistados, os entrevistados foram convidados a fornecer informações sobre seu trabalho e educação e porque estão interessados em UX.

Resultados

Relativamente aos resultados do estudo, na secção *background* dos entrevistados em que a maioria era da indústria e com formação na área de HCI, para a maioria deles o principal interesse em UX era projetar produtos melhores e viram UX como tema central para o seu trabalho profissional. Na secção "Declaração UX", os inquiridos entendem UX como dinâmica, dependente do contexto, e subjetiva, resultante de uma ampla gama de utilizadores e de potenciais benefícios que podem derivar de um produto. A UX é vista como algo novo, que deve ser uma parte do domínio de HCI e ser baseada em práticas UCD(User Centred Design).

Na secção "Definições UX", os entrevistados foram convidados a ler cinco definições de UX e comentar sobre eles. As cinco definições usadas no estudo foram:

1. Todos os aspetos da interação do utilizador final com a empresa, seus serviços e seus produtos. O primeiro requisito para uma experiência de utilizador exemplar é atender as necessidades exatas do cliente, sem barulho ou incomodar. Em seguida, vem a simplicidade e elegância que produzem produtos que fazem uma alegria de possuir, uma alegria de usar. A verdadeira experiência do utilizador vai muito além de dar aos clientes o que eles dizem que querem, ou fornecer lista de recursos. [9]
2. A UX é a consequência do estado interno do utilizador, as características do sistema projetado, e o contexto no qual a interação ocorre [10].
3. A UX é um conjunto de afetos que é provocada pela interação entre um utilizador e um produto [11],
4. A UX é o valor obtido a partir da interação com um produto ou serviço e do elenco de apoio no contexto de uso (por exemplo, tempo, localização e disposição do utilizado). [12]
5. A UX é a qualidade da experiência que uma pessoa tem ao interagir com um design específico [13].

Aparentemente, os entrevistados da indústria preferiram as definições 1 e 2 enquanto os académicos preferiram 2 e 3. Esta observação não é surpreendente dado que a definição 1 é da autoria das partes interessadas da indústria, enquanto 2 e 3 são de autoria dos académicos. Os entrevistados consideraram a qualidade de ser abrangente, fácil de entender, simples, clara, concisa e precisa como a definição mais relevante para uma definição UX. O que os entrevistados mais concordaram era que a experiência do utilizador deve ser avaliada enquanto este interage com um artefacto ao invés de após a interação. Por outro lado, acreditavam que experiências de utilização diferentes podem ser comparadas, apesar da natureza fugaz das condições que afetam a experiência do utilizador.

Um aspeto importante da pesquisa é perceber se a definição UX proposto pela norma ISO está em concordância com as opiniões recolhidas com o questionário e discutidas no

estudo. A definição da norma ISO diz que: *UX é a percepção de uma pessoa e as respostas que resultam do uso de um produto, sistema ou serviço*. A definição está em acordo com o ponto de vista da maioria dos entrevistados sobre a subjetividade do UX, e se concentra sobre as consequências imediatas do uso. De acordo com o ponto de vista da pesquisa a experiência do utilizador centra-se sobre a interação entre uma pessoa e algo que tem uma interface com o utilizador.

Em resumo, tendo em conta as conclusões do estudo, a UX relaciona-se directamente com interação de um utilizador com o sistema e deve ser avaliada enquanto o utilizador interage com o sistema e deve ser avaliada enquanto o utilizador interage com o sistema.

2.2 Eye-tracking e Leitura

Os sistemas de Eye-tracking são utilizados entre outros dispositivos, como técnica para gerar marcadores para a análise de leitura contínua [14]. Os movimentos dos olhos durante o processo de leitura têm sido extensivamente analisados, usando a metodologia eye-tracking para inferir a natureza dos processos cognitivos latente para a atividade leitura. Diferentes tipos de movimento dos olhos podem ser registados durante a leitura. As palavras são fixadas por cerca de 200-250 ms após o que ocorre uma sacada para a próxima palavra a ser fixada [15]. O movimento dos olhos durante a leitura depende tanto de variáveis linguísticas como oculares [16].

O modelo E-Z Reader [17] fornece um enquadramento teórico para entender como a identificação de palavras, o processamento visual, atenção e controle ocular determinam conjuntamente quando e para onde os olhos se movem durante a leitura. Qualquer discussão de modelos do controlo de movimento dos olhos deve começar com um breve panorama dos movimentos oculares durante a leitura, como se descreve na lista a seguir:

1. **Sacadas e fixações**, os olhos fazem movimentos curtos e rápidos chamado de sacadas que normalmente se movem para a frente cerca de 6-9 espaços de caracteres [15]. As sacadas leva 20-50 ms para completar, dependendo do comprimento do movimento, e virtualmente nenhuma informação visual é extraído durante os movimentos dos olhos. Entre as sacadas, os olhos permanecem fixos por breves períodos de tempo chamado fixações. A informação visual só é extraída durante as fixações, isso torna a leitura semelhante a uma apresentação de slides em que pequenos segmentos de textos são exibidos durante um quarto de segundo [18].
2. **Acuidade visual**, Uma das razões pelas quais os olhos estão em constante movimento na leitura é que existem limites severos para a quantidade de informação visual que pode ser processada durante a fixação [18]. A acuidade visual é máxima no centro da retina e diminui rapidamente para a periferia e finas discriminações visuais podem apenas ser feitas dentro da fovea.

3. **Aquisição de informações durante as fixações oculares**, Durante as sacadas, a visão é suprimida para que as informações necessárias para a leitura sejam adquiridas apenas durante as fixações [19].
4. **Onde fixar a seguir**, as decisões sobre onde se fixar próxima parecem ser determinadas em grande parte pelas pistas visuais de baixo nível no texto, tais como o comprimento da palavra e os espaços entre palavras. O comprimento de sacadas é influenciada pelo comprimento da palavra fixada e da palavra a direita da fixação.
5. **Quando mover os olhos**, A facilidade ou dificuldade relacionada com o processamento de uma palavra é influenciada principalmente quando os olhos se movem. Fatores linguísticos são principal determinante de quando mover os olhos. Vários estudos concluíram que os leitores olham mais para palavras de baixa frequência do que em palavras de alta frequência.

Para este projeto, os movimentos oculares foram utilizados para determinar o input de uma leitura com segmentação manual. Na segmentação manual o texto utilizado foi obtido a partir de um dispositivo de eye-tracking que captura as fixações e as sacadas realizadas durante a leitura de um texto contínuo. No final o texto original é segmentado de acordo com as sacadas naturais da leitura. Como veremos na secção 3.2 mais a frente.

2.3 Aplicações relacionadas

Esta sessão apresenta algumas aplicações relacionadas com a leitura, neste caso usadas por dispositivos móveis. Como exemplo vai ser apresentada dois leitores de texto, TEXT-Player Lite (Autoscroll), Reader Lite e Reading Player Lite.

2.3.1 TextPlayer Lite (Autoscroll)

TextPlayer [20] é um leitor de texto que permite otimizar textos longos, tais como livros para se adequar ao iPhone. O scroll é automático e permite que o texto se mova automaticamente, podendo ajustar a velocidade do deslocamento para se adequar a cada pessoa. Enquanto o scroll automático estiver ligado, a aplicação calcula o tempo restante assim como um leitor de música, este tempo é calculado com base na velocidade de deslocamento. A aplicação permite transferir ficheiros de texto do PC para iPhone através de web browser do PC, sem que seja necessária a instalação de algum software. Ficheiros de textos grandes podem ser carregados rapidamente através do processo de cache. Esta aplicação oferece um mínimo de esforço para ler textos longo no iPhone e facilita a leitura.

- **Funcionalidades**

De seguida são ilustradas as principais funcionalidades da aplicação TestPlayer(Autoscroll), e a figura 2.1 ilustra estas funcionalidades.

- Contém um scroll automático suave e um bom controlo de scroll
- Função Web-Server que permite transferir ficheiros de texto do PC para iPhone via Web browser do PC
- Função de cache de carregamento de ficheiro de texto
- Navegador de texto e mostra o tempo restante durante a rodagem do texto
- Permite configuração da fonte e da cor do texto

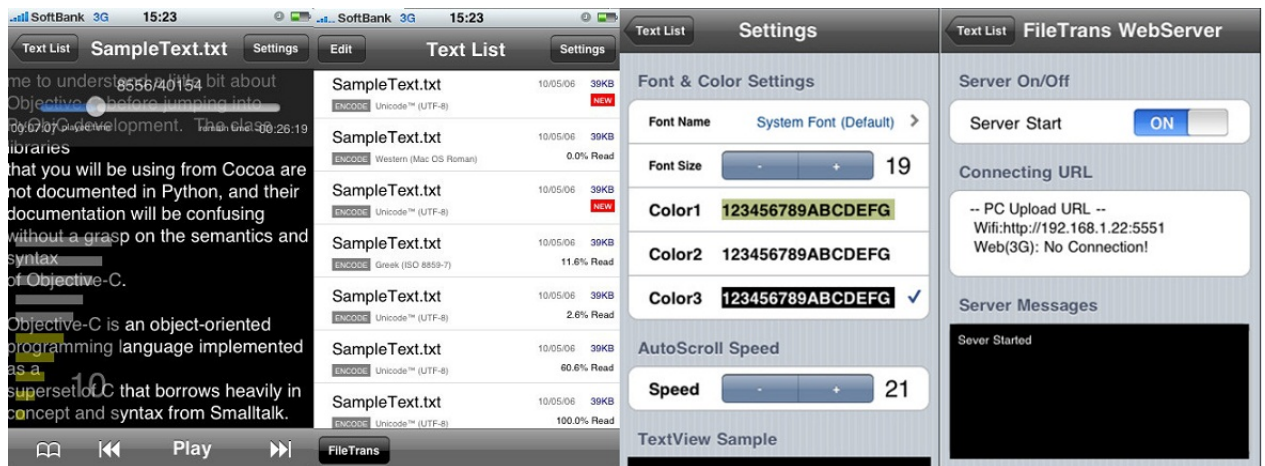


Figura 2.1: TextPlayer(Autoscroll)

2.3.2 Reader Lite

Esta aplicação [21] é um poderoso leitor de eBook para iPhone e iPod Touch. Este leitor permite que uma pessoa tenha todos os seus livros nas suas mãos. Foi desenvolvido do zero para dar uma experiência de leitura mais intuitiva possível.



Figura 2.2: Reader Lite

• Principais características

- Permite ao utilizador adicionar seus próprios livros e dá a opção de aceder uma biblioteca pública gratuita.
- Permite adicionar novos artigos, livros do computador diretamente na aplicação.
- Contém um dicionário para poder pesquisar as palavras.
- Pode-se ajustar o tamanho da fonte.

- Contém scroll automático que permite uma leitura livre de toque e também treina os olhos a ler num ritmo constante. Define-se uma velocidade de scroll automático e os livros fluem como nos filmes.
- As cores são escolhidas para conforto visual, assim podendo assim ler por várias horas.
- Tem um modo noturno embutido para tornar fácil ler livros de noite.
- Bookmarks rápidos que permite memorizar o lugar onde parou.

2.3.3 Reading Player Lite

Esta aplicação [22] permite que uma pessoa leia seus documentos, emails, etc usando o método de leitura flash. Em vez de percorrer o documento, o texto é lido em pedaços no mesmo local. Permite definir suas configurações para o comprimento do pedaço e texto, o tempo entre flashes. Pode-se também definir as configurações do nível de dificuldade. Esta é a configuração WPM (Words per minute) mais adaptada para a velocidade de leitura. Antes de dominar o Reading Player pode-se aumentar a WPM usando a função de treino.

- **Principais características do treino**

- 40 níveis de dificuldade, bloqueadas e tem que se desbloquear um por um;
- Mais de 5000 palavras e expressões únicas;
- Aumento gradual da WPM sobre os níveis;
- Pode reiniciar um nível para melhorar a pontuação.

- **Principais características**

- Fácil de guardar documentos;
- Três níveis de dificuldades;
- Define o WPM usando 3 parâmetros: tempo para mostrar o pedaço de texto, o tempo entre cada pedaço e o comprimento do pedaço;
- Muda o WPM durante a leitura do documento;
- Play/Pause/Stop da leitura;
- Retrocesso ou avanço rápido dentro do documento;
- Bookmarking, permite continuar de onde parou de ler;

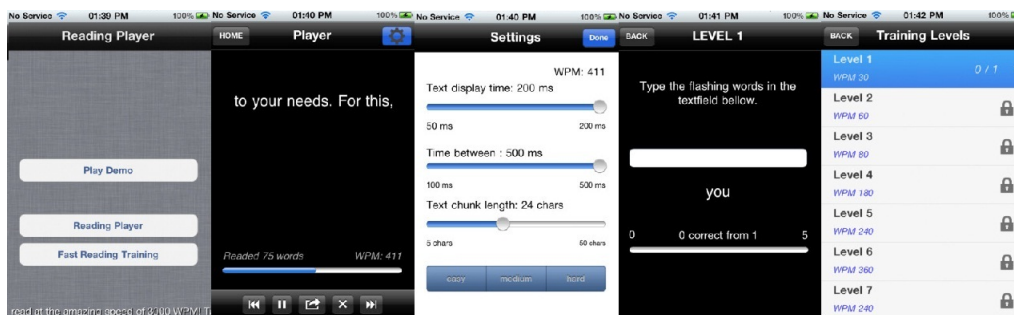


Figura 2.3: Reading Player Lite

2.4 A análise de leitura

A processo de leitura global tem sido analisado em vários estudos com a finalidade de compreender melhor como e porquê as pessoas leem um determinado texto. As pessoas podem ler de diferentes maneiras e para finalidades específicas de acordo com o tipo de informação que deseja extrair [4]. Os diferentes tipos de leitura:

Leitura recetiva, neste tipo de leitura a compreensão do texto exige alguma parte do texto já lido a ser mantido na memória de trabalho para permitir a integração do significado do texto a ser lido.

Leitura reflexiva, envolve interrupções por momentos de pensamento reflexivo sobre o conteúdo do texto.

Leitura superficial, leitura rápida que é usado para estabelecer uma ideia geral do texto. É útil quando é preciso verificar se o texto será útil para ler ou para determinar as partes que quer ler.

Exploração, se refere mais especificamente à procura do texto para ver se uma determinada parte de informações estão presente ou para localizar um pedaço de informação conhecido por estar no texto.

Leitura serial/ não serial, serial é quando o texto é lido de forma linear do início ao fim e não serial envolve movimento de uma secção de texto para outro.

Leitura simples ou repetida, um texto pode ser lido de uma só vez ou repetidamente.

Além destes diferentes tipos de leitura de texto, existem algumas ações de apoio associadas a leitura.

Sublinhado, É fácil e rápido, mas pode ser acompanhado de uma processamento profundo do texto quando precisa-se definir o que é importante a ser sublinhado.

Anotações, pode facilitar o nível mais profundo de compreensão do texto por causa da necessidade de definir o que é importante.

Outlining, Permite produzir uma representação alternativa do significado do texto.

Existem várias finalidades para a leitura de um texto. De seguida mostra-se os objeti-

vos de leitura mais comuns: **Ler para aprender**, tem a ver com a capacidade de aprender sobre algo, é uma das finalidades mais importante da leitura, é ser capaz de aprender sobre algo. **Ler para auto-informar**, refere-se a leitura que visa promover o conhecimento sobre um determinado tópico sem objetivo específico em que a informação vai ser aplicada. **Ler para procurar/responder perguntas**, Ao ler para pesquisa ou para responder a uma pergunta, é procura por informações no texto que satisfaz o objetivo da pesquisa. **Ler para a resolução de problemas**, envolve a leitura onde é preciso aceder a informações de múltiplas fontes. **Ler por prazer**, é caracterizada por alta-concentração e envolvimento emocional no texto, pode envolver a tentativa de de antecipar o que está a frente no texto. **Ler para aplicar**, refere-se a leitura quando instruções sobre como fazer algo estão presentes no texto. **Ler para discutir**, refere-se a leitura em preparação para uma aula ou uma reunião, envolve a revisão dos conceitos básicos para discussão. **Ler para resumir**, tem a finalidade de sintetizar um pedaço de texto para extrair pontos importantes.

2.5 Resumo

Em resumo, neste capítulo foram apresentados alguns conceitos de áreas relacionados com o trabalho desenvolvido e alguns exemplos de aplicações cujo conceito está relacionada com o trabalho. Começou-se por descrever um estudo que permite entender o conceito da UX, para entender melhor sobre o conceito da UX. Descreveu-se a utilização de eye-tracking no processo de leitura, incluindo a contribuição deste na realização deste projeto. Foram apresentadas alguns exemplos de aplicações relacionadas com o projeto: TextPlayer, que é um leitor de texto que permite otimizar a leitura de textos longos, tais como livros; Reader Lite, permite que uma pessoa tenha todos os seus livros nas mãos; Reading Player, que permite que uma pessoa leia seus documentos usando o método de leitura flash. Foram apresentados os conceitos e as funcionalidades de cada uma dessas aplicações. Por fim apresentou-se uma breve referêncnia a trabalhos que estudam a leitura numa perspectiva mais global.

Capítulo 3

Trabalho desenvolvido

Este capítulo descreve o trabalho realizado no âmbito do projeto, enumerando todas as fases do desenvolvimento do projeto.

3.1 Análise de Requisitos

A fase de análise de requisitos é a primeira atividade de um processo de desenvolvimento de software. A análise de requisitos possibilita ao desenvolvedor especificar a função e o desempenho do software e proporciona ao desenvolvedor e ao cliente os critérios para avaliar a qualidade logo que o software for construído, e também pode indicar a interação do software com outros elementos do sistema [23]. Uma compreensão completa dos requisitos dos sistemas é fundamental para um desenvolvimento de software bem-sucedido. Nesta secção a identificam-se os requisitos funcionais e não funcionais da aplicação assim como os casos de uso que permitem identificar o modo como o utilizador pode interagir com o sistema.

3.1.1 Requisitos funcionais

Os requisitos funcionais descrevem as funcionalidades que se esperam do sistema. No caso, para o TextReaderA foram identificadas as seguintes funcionalidades:

- **Configuração do modo de leitura**, que permite escolher o modo com que se pretende apresentar o texto de acordo com as diferentes formas de segmentação. As formas de segmentação são:
 1. Leitura contínua;
 2. Segmentação 1-1, palavra a palavra, em que cada palavra do texto é apresentado individualmente;
 3. Segmentação 3-3, palavra a palavra com apresentação das palavras adjacentes, esse estilo é inspirado pela consideração dos mecanismos de leitura para-

foveal, em que a apresentação de palavras seguintes no campo visual pode ter impacto na leitura geral;

4. Segmentação manual, conjuntos de palavras determinados pelos critérios externos. Permite segmentar um texto de acordo com critérios sintáticos, que determinam teoricamente os pontos de fixação visual num texto, ou de acordo com informação experimental, obtida a partir de dispositivos eye-tracking que capturam as fixações e sacadas efetivamente realizadas durante a leitura desse texto.
- **Configuração das fontes do texto**, permite alterar as fontes do texto, assim como a sua dimensão, de acordo com as preferências dos utilizadores.
 - **Configurar as cores do texto e do fundo**, o utilizador pode alterar tanto as cores de texto, como as cores do fundo de acordo com as suas preferências.
 - **Abrir o ficheiro de texto**, Para ler o texto o utilizador tem que escolher um ficheiro de texto num conjunto de textos disponibilizados.
 - **Ler o texto**, o utilizador pode ler um determinado texto, no modo de leitura que escolher.
 - **Avaliar a experiência de leitura do utilizador**. Esta funcionalidade permite avaliar a experiência de leitura de cada utilizador em relação ao texto lido, esta avaliação é feita no final de cada leitura em que o utilizador responde a um questionário sobre o texto lido e a sua experiência.
 - **Configurar a velocidade da apresentação do texto**, o utilizador pode alterar a velocidade da apresentação de texto. Esta funcionalidade está disponível no caso da leitura segmentada, em que o texto pode ser apresentado a diferentes velocidades.

3.1.2 Requisitos não funcionais

Os requisitos não funcionais descrevem as qualidades adicionais do sistema. Os requisitos não funcionais que foram identificadas para o TextReaderA foram os seguintes:

- **Desempenho**, está associado à eficiência, uso de recursos e tempo de respostas da aplicação. A aplicação TextReaderA deve responder dentro de um determinado intervalo.
- **Usabilidade**, está relacionado com a interação do utilizador com aplicação da aplicação. Define o nível de dificuldade que o utilizador terá para executar as operação. A aplicação TextReaderA deve apresentar uma interface que tenha uma curva de aprendizagem curta.

- **Adaptabilidade**, a aplicação deve adaptar-se facilmente ao ambiente que vai operar.

3.1.3 Casos de Uso

Os casos de uso têm como objetivo representar a interação entre um utilizador e um sistema. Permite ao cliente identificar as principais funcionalidades do sistema. Descreve o cenário que mostra as funcionalidades do sistema do ponto de vista do utilizador.

A Figura 3.1 representa o diagrama de Casos de Uso e demonstra as interações do utilizador com o sistema.

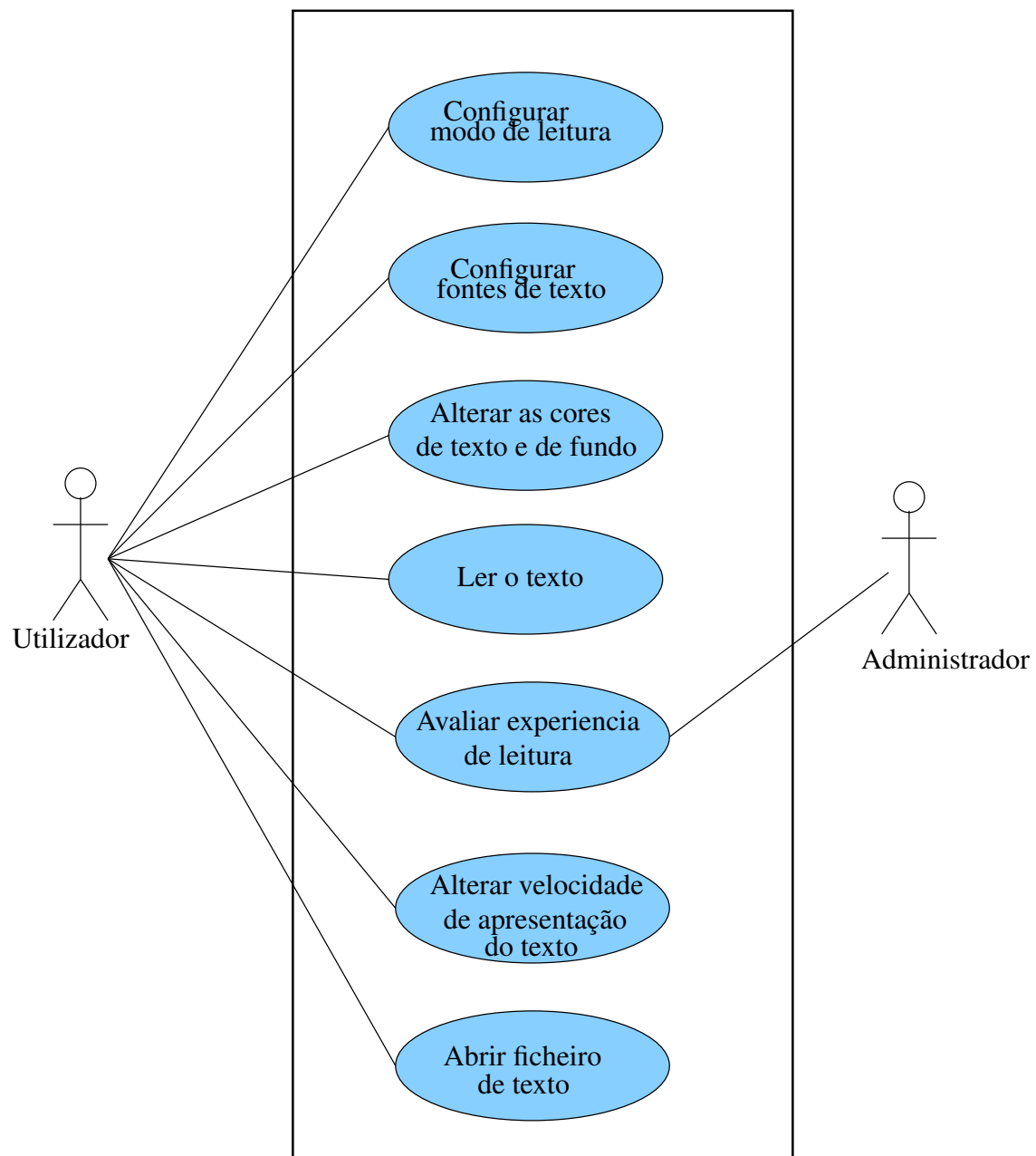


Figura 3.1: Casos de uso

3.2 Segmentação

Nesta secção são descritos as diferentes formas de segmentação de texto utilizados neste trabalho. Como já foi referido anteriormente foram utilizados 3 tipos de segmentação de texto: Segmentação 1-1, Segmentação 3-3 e Segmentação manual. De seguida apresenta-se uma breve descrição do modo como são feitas essas segmentações.

3.2.1 Segmentação 1-1

Na segmentação 1-1, ou palavra por palavra, cada palavra do texto é apresentado sequencialmente no centro do ecrã de acordo com a velocidade determinada.

3.2.2 Segmentação 3-3

A segmentação 3-3 é baseada na segmentação palavra por palavra mas neste caso cada palavra aparece acompanhado bilateralmente pela palavra anterior e seguinte do texto. A primeira palavra é acompanhado por(###). Esta segmentação é inspirada pela consideração dos mecanismos de leitura parafoveal em que a extração da informação visual à direita da palavra fixada pode ter impacto na leitura geral. Cada bloco de palavra é apresentado no centro do ecrã de acordo com a velocidade determinada.

3.2.3 Segmentação manual

A segmentação manual foi obtida a partir de dispositivos eye-tracking que capturam as fixações e sacadas efetivamente realizadas durante a leitura de um texto. O processo de captura das fixações será explicado de seguida.

Descrição detalhada

Os textos utilizados nas experiências (apêndice C) são textos curtos (70-100 palavras) em português, apresentados visualmente como uma imagem simples com uma fonte grande(16-20) e elevado espaçamento entre linhas. Esta configuração facilita as fixações e minimiza os requisitos de resolução de Eye Tracking.

Resultados do eye-tracking

Os dados dos movimentos oculares são guardados com o equipamento SMI [24] e são produzidos representações de sequências. A figura 3.2 ilustra uma das imagens obtidas.

Como mencionado acima, um conjunto de processamento de dados simples é realizado(em MATLAB [25]) para eliminar pequenas oscilações verticais ao longo de uma linha de texto, e para eliminar as sacadas inicial e final(ver movimento dos olhos a partir do centro da imagem do texto para o título do texto, e a partir do final do texto para o centro da imagem) Os resultados das várias observações são reunidas numa única imagem(como ilustrado na figura 3.3).

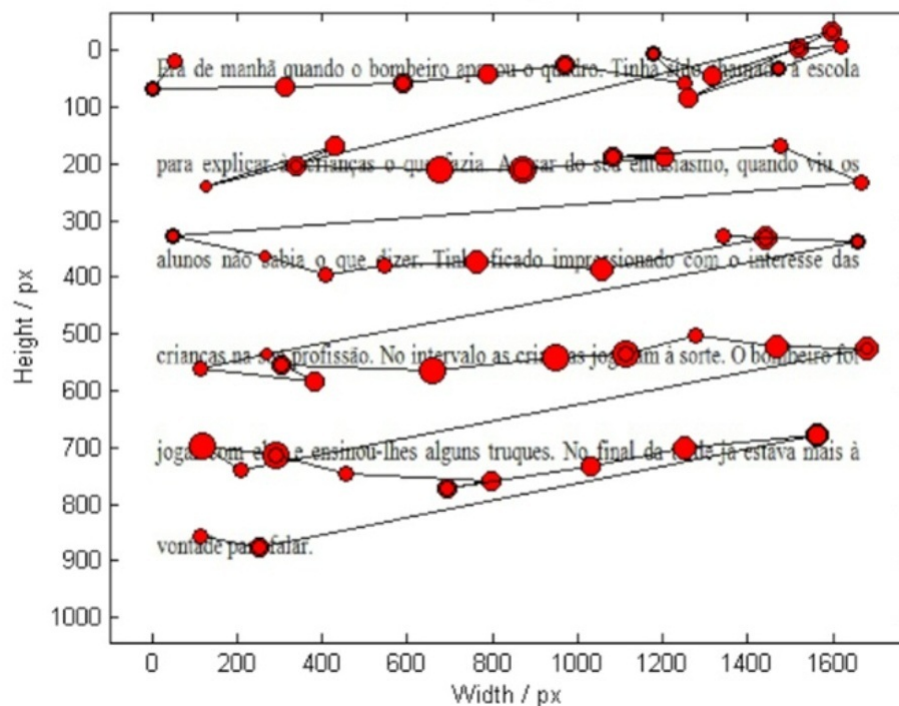


Figura 3.2: Imagem de texto com movimentos oculares

Segmentação manual

Com as imagens acima é construída uma tabela de segmentação com base na análise visual realizada por observadores independentes. Cada observador indica o segmento de texto que corresponde a cada fixação. A justificação para esse processo manual é a seguinte.

1. Como o objetivo é capturar a segmentação natural baseado em fixações, a utilização da técnica de área de interesse do eye-tracking retorna palavras simples(em vez de grupos de palavras), e um processo de agregação seria exigida.
2. O processo de segmentação manual é razoavelmente rápido, e o tempo necessário para a leitura do texto com uma pequena sobrecarga para a marcação de segmentos, e pode ser realizado por qualquer leitor disponível.
3. Os processos sintáticos e semânticos que subjacente a leitura original, também estão presentes na tarefa de segmentação, e uma análise automatizada baseado na técnica de processamento de linguagem natural é, portanto evitada.

Como seria de esperar, as segmentações propostas pelos vários observadores individuais não são únicos e a segmentação final é uma média, com base em decisões tomadas por um utilizador final. Tabela 3.2.3.

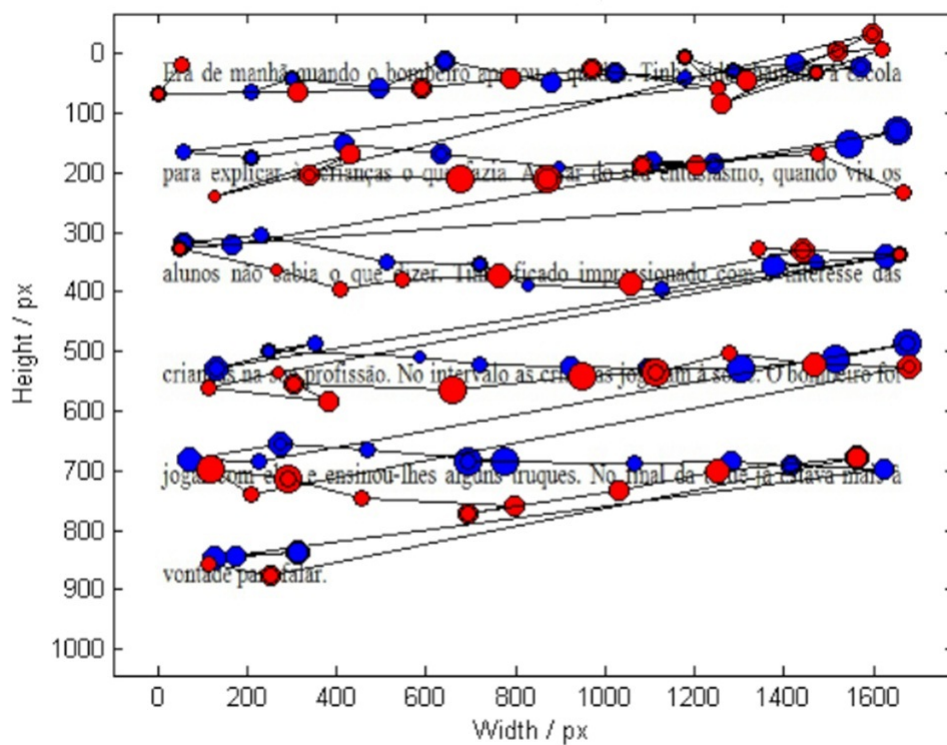


Figura 3.3: Combinação de imagens de movimento ocular de vários observadores

O processo de decisão é relativamente fácil e só se torna explícito quando não há consenso ou maioria entre as observações finais. No caso da amostra ilustrado abaixo, isso acontece em cerca de 17% das fixações, e mesmo assim, alguns dos casos indecisos diferem por uma única palavra de ligação, o que significa que apenas cerca de 11% das decisões implicam uma escolha na posição de uma palavra. A fusão de um conjunto de observações provavelmente vai levar a um pequeno aumento do número total de fixações. No caso do exemplo da tabela 3.2.3, o número total de segmentos é de 2% maior que o maior número de segmentos indicados pelos observadores.

Obs1	Obs2	Obs3	Segmentation decision
Era de manhã	Era de manhã	Era de manhã	Era de manhã
quando o policia	quando o	quando	quando
	policia	o policia	o policia
apagou o	apagou o	apagou	apagou o
quadro. Tinha	quadro.	quadro.	quadro
	Tinha	Tinha	Tinha
sido	sido	sido	sido
chamado à	chamado à	chamado	chamado à
escola para	escola	à escola	escola
explicar às	para explicar	para explicar	para explicar
crianças o	às crianças	às crianças	às crianças
que fazia.	o que fazia	o que fazia	o que fazia
Apesar do	Apesar do	Apesar	Apesar do
seu	seu	do seu entusiasmo	seu
entusiasmo	entusiasmo		entusiasmo
quando viu	quando viu	quando viu os	quando viu
os	os		os
alunos	alunos	alunos	alunos
não sabia o	não sabia	não sabia	não sabia
que dizer.	o que dizer.	o que dizer	o que dizer
Tinha ficado	Tinha	Tinha	Tinha
impressionado com	ficado impressionado	ficado	ficado impressionado
o		impressionado	
		com o	com o
interesse	com o interesse	interesse das	interesse
das	das		das
crianças na	crianças	crianças	crianças
sua	na sua	na sua profissão	na sua
profissão.No	profissão.		profissão.
intervalo as	No intervalo	No intervalo	No intervalo
crianças	as crianças	as crianças	as crianças
jogaram à	jogaram	jogaram à	jogaram à
sorte. O	à sorte.	sorte.	sorte.
bombeiro foi	O bombeiro	O bombeiro	O bombeiro
	foi	foi	foi
jogar com	jogar	jogar	jogar
elas e	com elas	com elas	com elas
ensinou-lhes	e ensinou-lhes	e ensinou-lhes	e ensinou-lhes
alguns	alguns	alguns truques	alguns
truques.	truques.		truques.
No final da	No final	No final	No final
tarde já estava	da tarde	da tarde	da tarde
	já estava	já estava	já estava
mais à	mais à	mais à	mais à
vontade para	vontade	vontade	vontade
falar.	para falar.	para falar.	para falar.

Tabela 3.1: Tabela de segmentação

A partir dos resultados da tabela, foi gerado um ficheiro XML com a segmentação resultante. Este texto é uma simplificação (mais curto) do texto utilizado na experiência descrita anteriormente.

```
1 <?xml version="1.0" encoding="UTF-8"?>
2 <texto>
3 <item>Era de manhã </item>
4 <item>quando o </item>
5 <item>policia </item>
6 <item>apagou o </item>
7 <item>quadro. </item>
8 <item>Foi à escola </item>
9 <item>para </item>
10 <item>contar </item>
11 <item>aos </item>
12 <item>alunos o </item>
13 <item>que fazia. </item>
14 <item>Apesar da </item>
15 <item>sua </item>
16 <item>calma </item>
17 <item>quando </item>
18 <item>viu os </item>
19 <item>alunos não </item>
20 <item>sabia o </item>
21 <item>que </item>
22 <item>dizer. </item>
23 <item>Ficou </item>
24 <item>sem palavras </item>
25 <item>com a </item>
26 <item>atenção dos </item>
27 <item>alunos. </item>
28 <item>No fim </item>
29 <item>das aulas </item>
30 <item>o policia </item>
31 <item>foi </item>
32 <item>brincar com eles </item>
33 <item>e contou </item>
34 <item>algumas </item>
35 <item>coisas </item>
36 <item>que </item>
37 <item>fazia. </item>
38 <item>No final da </item>
39 <item>tarde já estava </item>
40 <item>mais calmo </item>
41 <item>para </item>
42 <item>falar </item>
43 </texto>
```

Listagem 3.1: Ficheiro XML com a segmentação manual

3.3 Implementação

3.3.1 Ambiente de programação

Esta secção descreve as linguagens de programação e as ferramentas utilizadas na realização do projeto.

Para a implementação da camada da interface com o utilizador, utiliza-se a linguagem de programação JAVA, os componentes Swing [26] e a ferramenta Eclipse [27]. A escolha da linguagem JAVA e dos componentes Swing foi determinada pelo facto de a linguagem JAVA ser uma linguagem de corre em qualquer sistema operativo, está rodeada de API's eficazes e completas, baseia-se numa arquitetura aberta e possui uma interface gráfica (Swing), permitindo criar aplicações complexas e de resposta rápida.

Algumas vantagens [28] da utilização da ferramenta Eclipse são: É um software gratuito, sendo possível desenvolver aplicações em várias linguagens, por exemplo Java, PHP, C/C++ e muitos mais. Permite um rápido desenvolvimento, cria geradores de interfaces com o utilizador (Java Swing, Web, etc.). O ambiente Eclipse precisa apenas de uma máquina virtual JAVA para funcionar, por isso pode ser executado em qualquer sistema operativo.

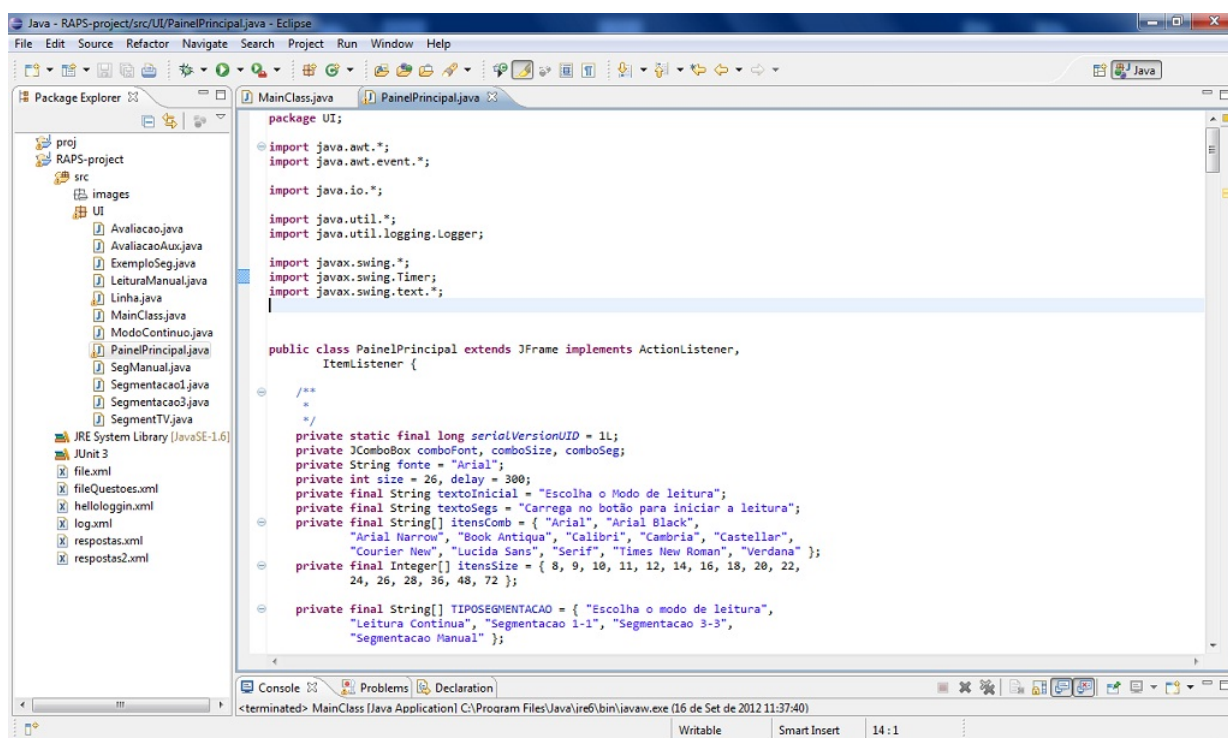


Figura 3.4: Ambiente de desenvolvimento da aplicação TextReaderA

3.3.2 Interface para sistemas externos

O sistema TextReaderA tem mecanismos básicos de ligação a sistemas externos. A finalidade destas interfaces é mediar a comunicação entre a aplicação e outros sistemas, e registar dados sobre as várias sessões de leitura. A aplicação TextReaderA utiliza XML para importar/exportar dados de /para outros sistemas, por exemplo os dados do eye-tracking utilizados na segmentação manual são obtidos a partir de ficheiro XML. O questionário utilizado para avaliar a experiênciadados utilizadores também é disponibilizado através de ficheiros XML D.1. Os resultados das experiências também são registados nestes ficheiros. Para os ficheiros que foram criados manualmente foi utilizado o tutorial da [29] e implementação para ler estes ficheiros a partir de código Java foi utilizado a API java para XML [30].

```
1 <?xml version="1.0" encoding="UTF-8"?>
2 <questionarios>
3     <questao id="1">Gostou de ler o texto?</questao>
4     <questao id="2">A história é fácil de ler? </questao>
5     <questao id="3">O Texto é fácil de seguir?</questao>
6     <questao id="4">Percebeu o sentido da história?</questao>
7     <questao id="5">Recorda de algum personagem do texto?</
    questao>
8     <questao id="6">É fácil entender o conceito da aplicação?</
    questao>
9     <questao id="7">É fácil controlar e usar a aplicação? </
    questao>
10    <questao id="8">Seria útil para usar em dispositivos móveis?
    </questao>
11 </questionarios>
```

Listagem 3.2: Ficheiro XML com o questionário utilizado na aplicação

3.3.3 TextReaderA

Esta secção tem como finalidade documentar toda a implementação das funcionalidades enumeradas na análise de requisitos. Como já foi referido anteriormente o projeto foi implementado utilizando a linguagem de programação JAVA e as componentes SWING. Esta implementação consiste em criar uma aplicação em que vão ser integradas as opções que serve de interface com o utilizador.

Todas as funcionalidades da aplicação foram implementadas utilizando os tutoriais disponíveis em [31] e [32]. Para implementar este projeto foram criadas seis classes, documentadas abaixo.

A classe MainClass.java é a classe principal que tem como finalidade criar e visualizar a janela principal, criada a partir da classe PainelPrincipal.java que é a classe que serve de base para o desenvolvimento de toda a aplicação. Como mostra a figura 3.5, as funções que o utilizador pode manipular são: **1- Escolher o ficheiro de texto, 2- Configurar**

o modo de apresentação de texto, 3- Configurar a fonte e o tamanho das letras, 5- Configurar a velocidade de apresentação de texto, 6- Configurar as cores do texto e do fundo do ecrã e 7- Avaliar a experiência do utilizador.

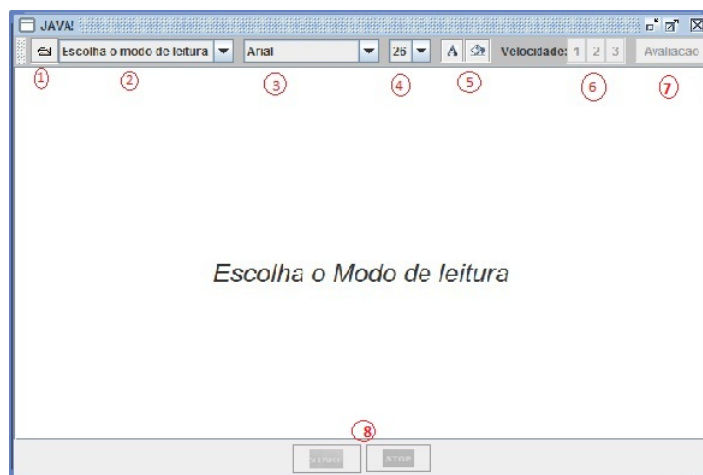


Figura 3.5: TextReaderA (janela principal)

- **Abrir o ficheiro de texto**

Os textos utilizados na experiência são guardados nos ficheiros de texto e antes de prosseguir para a leitura do texto o utilizador tem que abrir o ficheiro onde se encontra o texto a ser apresentado. Esta operação foi implementada utilizando a componente JFileChooser que permite ao utilizador escolher um ficheiro numa diretoria. O método `showOpenDialog()` permite ao utilizador navegar nas pastas e escolher o ficheiro de texto, e o método `getSelectedFile()`, que devolve o nome do ficheiro selecionado. Este ficheiro depois é aberto e o texto lido é guardado numa string.

- **Escolher o modo de apresentação de texto**

Esta opção o utilizador pode escolher o modo de apresentação de texto de acordo com os diferentes tipos de apresentação disponibilizados através do JComboBox. E o painel de texto é atualizado com o texto escolhido e com o modo de leitura definida. Para fazer esta atualização é utilizado a StyledDocument com as suas configurações e o texto é adicionado através do método `insertString()`.

Os modos de apresentação de texto podem ser:

- **Leitura contínua**, como o texto é apresentado continuamente no ecrã, neste caso o texto lido é guardado como uma única string e depois é inserida no documento com as configurações pretendidas pelo utilizador e depois é inserida no painel principal como foi explicado anteriormente.

A figura 3.6 mostra a visualização do painel quando for escolhida o modo de leitura contínua.

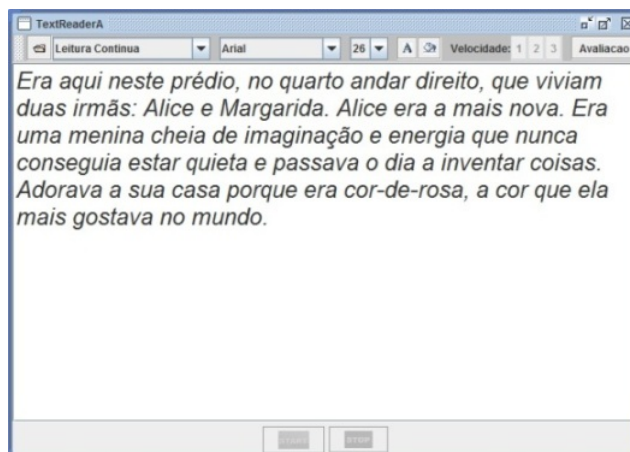


Figura 3.6: TextReaderA (Leitura contínua)

- **Leitura Segmentada**, neste caso o texto todo guardado numa string é separado em vários segmentos, conforme a segmentação escolhida. Cada segmento de texto é introduzido individualmente no painel, a inserção desses segmentos de texto é controlada através de um *Timer* em que o determina o tempo que separa a inserção de cada segmento.

Para a **segmentação 1-1** em que cada palavra do texto é apresentado no centro do ecrã, o texto lido é dividido em palavras individuais através do método `split()`, estas palavras é depois guardadas num array de string e inserida individualmente no ecrã. Ver figura 3.7.

Na **segmentação 3-3** o procedimento é igual a segmentação 1-1 na altura de dividir o texto, a diferença é que cada palavra do texto é inserida juntamente com a palavra anterior e a seguinte do texto e pós-procuradas manualmente(ver 3.2 adiante). Figura 3.8

Na **segmentação manual** a segmentação é obtida a partir de dispositivos eye-tracking que capturam as fixações e as sacadas realizadas durante a leitura de um texto. Estes segmentos de texto é guardado num ficheiro XML. O ficheiro XML é lido através do `DocumentBuilder` e de um parser do ficheiro, sendo cada segmento de texto introduzida no painel da mesma forma que as outras formas de segmentação. Figura 3.9

- **Configurar a fonte e o tamanho das letras**, para estas opções foi utilizado `JComboBox` com as opções disponíveis, e fonte e o tamanho do texto é alterado com as opções selecionadas.
- **Configurar a velocidade de apresentação de texto**. Nos diferentes tipos de lei-

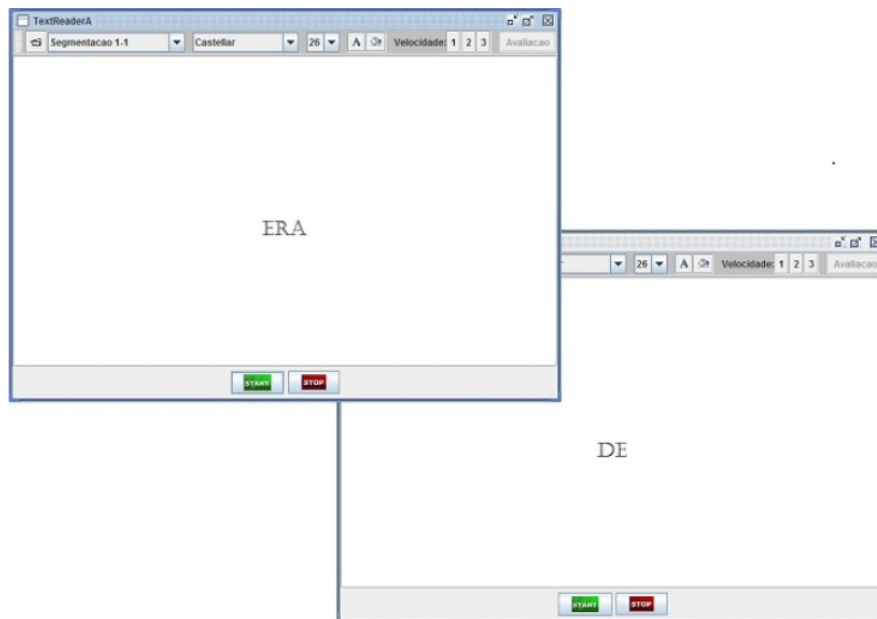


Figura 3.7: TextReaderA (Leitura com segmentação 1-1)



Figura 3.8: TextReaderA (Leitura com segmentação 3-3)

tura segmentada a velocidade de apresentação pode ser definida de acordo com as preferências do utilizador. Foram definidos 3 velocidades (Lento, normal e rápido), a partir de 3 botões. Foi determinado um valor para cada tipo de velocidade. É definido um delay(atraso) de 300ms para a velocidade mais rápida e é nesta velocidade que começa a apresentação de texto de qualquer uma da segmentações. O delay das outras velocidades é obtido multiplicando por 2 para a velocidade 2 e por 4 para a velocidade 1(mais lenta).

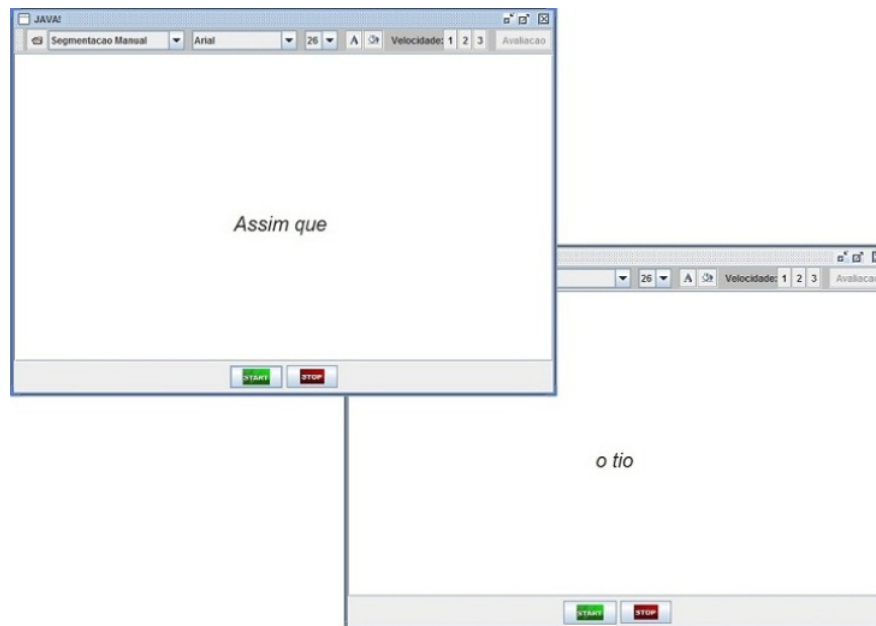


Figura 3.9: TextReaderA (Leitura com segmentação manual)

- **Configurar as cores do fundo e das letras**, para esta opção utiliza-se JButoon para criar os botões e é utilizado JColorChooser com a paleta das cores. A cor escolhida é devolvida através de um objeto do tipo Color e é usado para alterar a cor do fundo com o método setBackground(Color) e a cor do texto com o setForeground(Color).
- **Avaliar a experiência de leitura**, no final de cada leitura o utilizador responde a um conjunto de questões que procura avaliar a sua experiência de leitura e a usabilidade da aplicação. Para implementar esta funcionalidade primeiro foi criado um ficheiro XML com as questões que são usadas na avaliação. Este ficheiro depois é lido e é criado um painel onde o utilizador pode responder a um conjunto de questões e submeter a resposta. As respostas do utilizador é guardado noutro ficheiro XML.

3.4 Concretização

Nesta secção são apresentados as descrições das classes criadas para a implementação do projecto. Para este projeto foram implementadas sete classes e passo para a descrição das mesmas de seguida:

- **MainClass.java**

Esta classe contém o método `main` que é o método principal da aplicação, que determina o início da execução da aplicação. No método `main` utiliza-se o método `invokeLater()` do `SwingUtilities`, este por sua vez chama o método `createAndShowGUI()` da classe `PainelPrincipal`, que passo a descrever de seguida.

- **PainelPrincipal.java**

Esta classe estende a classe `JFrame` e contém praticamente toda a implementação da aplicação, nela contém a criação do painel principal, dos componentes que compõem a aplicação, como botões, menus, etc. Também contém as ações que dos dos componentes da aplicação. Pode-se encontrar os seguintes métodos:

- **inicializar()**, que permite inicializar todos os componentes que fazem parte da aplicação, inicializa as opções de apresentação do texto, por exemplo o alinhamento vertical e horizontal, a fonte, etc; inicializa a `toolBar` onde se pode encontrar os menus, o painel dos botões, o painel onde vai ser apresentado o texto (`textPane`) e no fim adiciona todos os componentes no Painel principal.
- **criarBotoes()**, cria e inicializa todos os botões da aplicação, adiciona a ação de cada um e adiciona-os nos respetivos painéis.
- **criarComboBoxes()**, permite criar todos *ComboBoxes* que são usados na aplicação, a partir da classe `JComboBox`, adiciona-os no painel dos menus e adiciona um *listener* a cada um. Estes são utilizados para mostrar as opções de leitura, as fontes e os tamanhos do texto.
- **lerTxt(File file)**, lê um ficheiro de texto passado como parâmetro, utiliza um objeto da classe `Scanner` para ler o ficheiro e guarda o texto lido numa string.
- **actionPerformed(ActionEvent e)**, este método é chamado quando acontece algum evento quer dos botões ou dos *combo boxes*, cada vez que carregar num botão ou escolher-se um item do combobox este método é chamada e compara o evento recebido no `ActionEvent` e compara com os componentes e executa as opções pedidas.
- **setPainelTexto(String seg)**, este método atualiza o painel de texto de acordo com o tipo de segmentação de texto passado, este chama o método `lerTxt()` para ler os textos e depois chama a classe específica para cada tipo de segmentação

e inicializa um *Timer* (temporizador) que dependendo da velocidade escolhida determina um *delay*(atraso) e chama o *listener* da classe escolhida e adiciona o documento devolvido de cada classe ao *textPane*.

- **startTimer()**, inicializa o temporizador através o método *start()* da classe *Timer*.
 - **stopTimer()** pára o temporizador através do método *stop()* da classe *Timer*
 - **createAndShowGUI()**, este método é responsável pela inicialização dos componentes da aplicação, começa por criar a janela principal a partir de um objeto da classe *PainelPrincipal()*, determina as dimensões da janela, chama o método *inicializar()* mencionado anteriormente para inicializar os componentes da aplicação e por fim chama o método *setVisible(true)* para visualizar a janela.
- **LeituraContinua.java**, esta classe trata do modo de leitura contínua. Recebe como parâmetro do construtor a string com o texto selecionado e chama o método *tratarTexto()*. Este método cria um documento a partir da classe *StyledDocument* e adiciona o texto no documento criado a partir do método *insertString()* desta classe. Também contém o método *getDocument()* que devolve o *StyledDocument* com o documento criado.
 - **Segmentacao1.java**, esta classe trata da implementação da segmentação 1-1, recebe como parâmetro a String com o texto lido e pega neste texto e divide em palavras individuais utilizando o método *split()* da classe *String* e é guardado num array de string. Nela contém o método *tratarTexto()* que devolve um listener que depois vai ser passado no argumento do *Timer* que foi referido anteriormente que também recebe um *delay*(atraso) que determina o período pelo qual é executado a ação do método. Por exemplo se for escolhida a velocidade 3 (ou rápida) que é a escolhida por omissão, o método é executado a cada 300ms, ou seja, de 300 em 300 ms uma palavra é introduzida no ecrã. Esta introdução é feita utilizando o método *insertString()* e *remove()*. Antes de fazer a inserção faz-se a verificação se corresponde a primeira primeira palavra do texto ou nao. Se for a primeira palavra do texto chama-se o método *insertString()*, passando a string com a palavra obtida de um dado índice do array, que adiciona a palavra no documento. Caso contrário chama-se primeiro o método *remove()* para remover a palavra que lá está para depois chamar *insertString()* para inserir outra palavra.

No final calcula-se o tempo de leitura através de *System.currentTimeMillis()* e abre a janela de avaliação que permite ao utilizador avaliar a sua experiência que vai ser descrita a frente.

- **Segmentacao3.java**, Nesta classe o procedimento é idêntico ao anterior, a diferença

está na hora de inserir o texto no ecrã. Neste caso, antes de inserir, criamos uma string em que vai ser guardada o segmento de texto a ser introduzido, o primeiro segmento de texto criado juntado "####" com a primeira e segunda palavra do texto, os restantes segmentos é criado sempre com a palavra do índice anterior, do índice actual e do posterior.

- **SegManual.java**, esta classe serve para implementar a segmentação manual. Recebe como parâmetro o ficheiro de texto escolhido, depois é chamado o método `loadXML()`, que lê o ficheiro em xml correspondente ao ficheiro escolhido, cria um `DocumentBuilder` para guardar o texto lido a partir do método `parse()`, depois é usado o `getElementByTagName()` com o item onde estão guardados cada segmento de texto e guarda cada segmento numa lista. O método `tratarTexto()` funciona da mesma forma que das outras classes de segmentação explicadas anteriormente.
- **Avaliacao.java**, esta classe trata do painel da avaliação das experiências e de guardar os resultados dos mesmos. Está dividida pelos seguintes métodos:
 - **loadXML()**, este método faz o carregamento do ficheiro XML com as questões que são usados na avaliação e cria um painel com as questões. Primeiro chama o método `parseXML()` que faz o *parse* do ficheiro citado anteriormente e devolve um `Document`, e através do método `getElementByTagName` vai buscar as questões, depois cria um `JPainel` e coloca as questões e na frente de cada questão cria-se um `RadioButton` onde são colocados as opções de escolha que está numa escala de 1-5.
 - **parseXML(String file)**, este método devolve o documento com o resultado do parse do ficheiro XML passado como parâmetro. Cria um `DocumentBuilderFactory` que permite criar um `DocumentBuilder`, que faz o parse do ficheiro e guarda num `Document`.
 - **criarResultado()**, este método é responsável pela criação o ficheiro onde são guardados as respostas do questionário. Primeiro começa-se por criar uma instância de `DocumentBuilderFactory` e usa-se o método `newDocumentBuilder()` para criar um `DocumentBuilder` e cria um objecto para criar uma árvore com `Document`. Utiliza o método `createElement()` para criar um objecto do tipo `Element` que será tag raiz do documento que será "Resultados", este método também é usado para criar as outras tags os que compõem o documento, usa o método `appendChild()` para adicionar as tags no documento e o método `setTextContent()` para adicionar valores aos tags criados. No final chama o método `converter()` para converter o documento criado no formato XML.
 - **converter()**, este método recebe como parâmetro um objecto do tipo `Document` e converte numa string em formato XML. Começa-se por criar uma instância

de Transformer e usa o método `setOutputProperty()` que cria a formatação do XML no ficheiro. Cria-se um `DOMSource` com o documento recebido, cria uma `StreamResult` e usa o método `transform()` para processar a árvore do `DOMSource` para um `StreamResult`, depois usa `getWriter` do `StreamResult` para recuperar o conteúdo em formato de uma string e usa `FileOutputStream` para guardar o resultado.

- **`createAndShowGUI()`**, este método define o tamanho e a localização da janela e chama o método `setVisible` para visualizar a janela.

Alguns exemplos do descrito pode ser encontrado em anexo no apêndice D.

3.5 Plano de desenvolvimento

Nesta secção é apresentado o plano do projeto, onde são enumeradas as tarefas necessárias para levar a cabo o projeto, bem como os recursos e prazos envolvidos. De seguida faz-se a calendarização do projeto a partir de um mapa de Gantt que descreve o fluxo de tarefas e os prazos envolvidos na realização de cada tarefa especificada. As tarefas necessárias para a realização do projeto:

- Outubro 2011
 - Análise do problema
 - Pesquisa de sistemas e ambientes relacionados,
- Novembro 2011 a Janeiro 2012
 - Especificação do ambiente pretendido, ferramentas externas e módulos a desenvolver
 - Identificação e especificação dos requisitos
 - Escrita do relatório preliminar
- Dezembro 2011 a Abril 2012
 - Implementação da solução
 - Documentação das funcionalidade implementadas
- Abril 2012 a Julho 2012
 - Preparação de ambiente de testes
 - Criação de plano de teste
 - Execução dos testes com utilizadores
 - Documentação dos resultados dos testes
- Julho 2012 a Setembro 2012
 - Escrita do Relatório final

O mapa de Gantt referente ao plano de desenvolvimento deste trabalho encontra-se em anexo no apêndice A.

Capítulo 4

Experiências e Resultados

Este capítulo apresenta as experiências realizadas no trabalho. São descritas as várias experiências feitas com utilizadores e uma análise dos resultados obtidos. O objetivo destas experiências é recolher as opiniões dos utilizadores em relação à aplicação e principalmente em relação à experiência de leitura.

4.1 Descrição das experiências

Um total de 15 utilizadores participaram nas experiências. Todos são estudantes do ensino superior com a faixa etária do 20-34 anos, a maioria dos participantes eram estudantes da área de informática. Todos os testes foram realizados no laboratório do LaSIGE. Foram utilizados quatro textos diferentes e lidos em 4 maneiras diferentes.

Nestas experiências o cenário é o seguinte:

- O sujeito é apresentado à ferramenta efetuando um breve teste das funcionalidades.
- O sujeito lê sucessivamente quatro textos diferentes, usando as várias formas de segmentação referidas (leitura contínua, segmentação 1-1, segmentação 3-3, segmentação manual).
- No fim de cada passo responde a um questionário simples mostrado de seguida.

4.1.1 Questionário utilizado na avaliação

As questões que fazem parte do questionário utilizado na avaliação da experiência do utilizador são os seguintes:

1. Gostou de ler o texto?
2. A história é fácil de ler?

3. O texto é fácil de seguir?
4. Percebeu o sentido da história?
5. Recorda de algum personagem da texto?
6. É fácil entender o conceito da aplicação?
7. É fácil controlar e usar a aplicação?
8. Seria útil para usar em dispositivos móveis?

No final são registadas as várias configurações de apresentação (fonte, tamanho de letra, etc.), bem como a duração da leitura.

4.2 Análise de Resultados

Nesta secção apresentam-se os resultados das experiências assim como a análise desses resultados. As tabelas 4.1, 4.2, 4.3, 4.4 mostram os resultados em bruto das experiências realizadas. Cada utilizador tinha que ler quatro textos das quatro formas possíveis e no fim responder ao questionário apresentado anteriormente. O questionário tinha de ser respondido com base na escala de Linkert [33], indicando a sua opinião em relação a sua leitura e a qualidade da aplicação, com excepção da questão Q5 em que tinha que indicar o nome de um personagem para perceber se estava atento a leitura e se percebeu o texto lido. São registadas as respostas do questionário de cada utilizador, o tempo que levou a ler, o tipo e o tamanho da fonte utilizada.

• **Experiência 1, Leitura Contínua**

User	Q1	Q2	Q3	Q4	Q5	Q6	Q7	Q8	Time(s)	Fonte	Size
1	5	5	5	5	Alice e Margarida	4	4	3	58	Arial	26
2	5	5	5	4	Vizinha	4	5	4	47	Arial	26
3	5	4	5	5	Tio	4	4	4	74	Arial	26
4	3	4	4	4		3	3	4	32	Calibri	36
5	5	5	5	4	polícia	4	4	4	40	Arial	26
6	5	5	5	5	Alice e Margarida	4	4	5	26	Arial	26
7	3	4	5	5	tio,mãe, filho	5	5	4	75	Arial	26
8	3	2	3	4	polícia	4	4	4	40	Arial	26
9	4	5	4	4	Margarida	4	4	4	64	Arial	26
10	4	4	4	5	Alice	4	4	3	42	Arial	26
11	4	4	4	3	Tio	4	4	3	16	Arial	26
12	4	5	5	5	pintor	5	5	4	40	Arial	26
13	4	5	5	5	tio	4	5	5	27	Arial	26
14	3	4	4	4	pintor	4	4	3	19	Arial	26
15	5	5	5	5	polícia	5	5	5	38	Arial	26
Média	4,1	4,4	4,5	4,1	93%	4,1	4,3	3,9	42,5		26,7

Tabela 4.1: Resultados da Experiência 1 (Leitura Contínua)

Comentário 1: Com base nos resultados da tabela 4.1 verificou-se que nesta primeira experiência os utilizadores quase não exploraram as funcionalidades da aplicação, preocupando-se com a leitura que estão a fazer, visto que foi o primeiro contacto que tiveram com aplicação. Indica que, com a leitura contínua, os utilizadores mantêm a tipo e tamanho da fonte, e o tempo médio de leitura é de 42,5 segundos. O tamanho médio das letras foi de 26.7. Em relação à opinião dos utilizadores sobre o texto lido (Q1-Q4), numa escala de 1 a 5, a maioria dos utilizadores consideraram que os textos apresentados eram fáceis de seguir(4,13), que a história era fácil de ler(4,53), que gostaram de ler o texto (4,13) e que perceberam o sentido da história(4,47). Pode-se afirmar que a experiência de utilizador nesta experiência foi positiva visto que todas as questões apresentaram uma média acima de 4.

• **Experiência 2, Leitura com segmentação 1-1**

User	Q1	Q2	Q3	Q4	Q5	Q6	Q7	Q8	Time(s)	Fonte	Size
1	4	5	4	3	pintor	3	5	4	32	Arial	26
2	5	5	5	4	Tio	4	5	4	49	Arial	26
3	5	5	5	5	pimtor	3	5	4	24	Arial Black	12
4	3	3	3	4	policia	4	3	4	30	Calibri	36
5	5	5	5	5	Alice	3	4	4	71	Arial	26
6	3	4	3	2		5	4	5	61	Arial	48
7	4	5	4	5	senhora, pintor	5	5	5	32	Courier New	48
8	5	5	5	5	Alice	3	4	4	38	Book Antiqua	26
9	3	4	3	5	pintor	3	4	4	26	Arial	26
10	5	5	5	5	policia	3	4	4	68	Arial	26
11	3	4	3	3	Policia	4	4	3	35	Times New Roman	28
12	1	4	2	2	tio	5	5	4	63	Arial	26
13	5	5	4	5	meninos	5	5	5	39	Arial	26
14	3	3	3	4	tio	4	4	4	53	Arial Narrow	20
15	5	5	4	5	tio, mãe, filho	4	5	5	56	Arial	26
Média	3,9	4,5	3,9	4,1	93%	3,9	4,4	4,2	45,1		28,4

Tabela 4.2: Resultados da Experiência 2 (Segmentação 1-1)

Comentário 2: Com o resultado da tabela 4.2, verificou-se que nesta experiência os utilizadores começaram a ficar mais à vontade com a aplicação e começaram a explorar mais funcionalidades, por exemplo a fonte e o tamanho do texto, o tamanho médio das letras do texto foi de 28,4. Entretanto o tempo médio de leitura foi de 45,1 segundos, neste caso os utilizadores tinham a possibilidade de alterar a velocidade de apresentação do texto, a maioria dos utilizadores optaram pela velocidade mais baixa para poderem controlar melhor a leitura. A opinião dos utilizadores nesta experiência variou, os utilizadores não gostaram tanto de ler o texto e acharam que o texto era fácil de seguir nestas condições, ambos com um valor abaixo de 4, ou seja 3,9. Entretanto acharam que o texto era fácil de ler(4,5) e que perceberam o seu sentido (4,1).

• **Experiência 3, Leitura com segmentação 3-3**

User	Q1	Q2	Q3	Q4	Q5	Q6	Q7	Q8	Time(s)	Fonte	Size
1	4	4	4	4	Tio	3	4	4	82	Arial	26
2	4	4	3	3	Polícia	3	5	4	48	Arial	26
3	5	5	5	5	polícia	4	3	4	43	Arial Black	12
4	3	3	4	3	polícia	5	3	3	47	Calibri	36
5	4	5	3	4	pintor	4	4	4	93	Arial	26
6	4	5	4	5	Tio	4	4	5	119	Arial	48
7	1	3	5	3	polícia, alunos	5	5	4	55	Arial	26
8	4	5	3	4	tio	4	4	4	53	Arial	26
9	3	5	4	5	tio	3	3	4	23	Arial	26
10	4	4	3	4	tio	4	4	4	47	Arial	26
11	3	3	2	2	Alice	4	4	3	45	Times New Roman	28
12	2	3	3	1	polícia	5	5	3	107	Arial	26
13	4	5	5	5	Margarida	5	5	5	66	Book Antiqua	26
14	2	2	2	3	polícia	4	3	4	42	Castellar	20
15	5	4	4	4	pintor, senhora	5	4	5	53	Arial	26
Média	3,5	4	3,6	3,7	100%	4,1	4	4	61,5		26,9

Tabela 4.3: Resultados da Experiência 3 (Segmentação 3-3)

Comentário 3, os resultados de experiência são ilustrados na tabela 4.3. Nesta experiência os utilizadores continuaram a explorar as diversas funcionalidades da aplicação, variaram a fonte e o seu tamanho, o tamanho médio das letras do texto foi 26,9. Nesta experiência também os utilizadores tinham a possibilidade de alterar a velocidade de apresentação de texto, e tal como a segmentação 1-1 a maioria optou por utilizar a velocidade mais baixa, isto nota-se devido ao tempo médio de leitura que apresenta valores muito superior as outras experiências. O tempo médio nesta experiência foi de 61,5 segundos. Em relação à experiência de leitura dos utilizadores, não gostaram de muito de ler e tiveram dificuldades de seguir o texto, e muitos não conseguiu entender o sentido da história.

• **Experiência 4, Leitura com segmentação manual**

User	Q1	Q2	Q3	Q4	Q5	Q6	Q7	Q8	TL(s)	Fonte	Size
1	4	4	4	4	policia	4	4	4	31	Arial	26
2	5	5	4	3	Margarida	3	5	4	23	Arial	26
3	5	5	5	5	Margarida	5	5	4	11	Arial Black	12
4	3	4	4	3	Tio	3	4	4	10	Calibri	36
5	5	5	5	5	tio	4	4	4	39	Arial	26
6	5	4	4	4	Policia	4	4	5	35	Arial	48
7	5	5	5	5	alice, margarida	4	4	4	26	Courier New	48
8	5	5	5	5	pintor	4	4	4	39	Arial	26
9	4	4	4	4	policia	3	3	4	32	Courier New	28
10	5	5	5	5	pintor	4	4	4	41	Arial	28
11	3	3	3	4	pintor	4	4	4	38	Courier New	22
12	5	5	5	5	Alice	5	5	3	53	Arial	26
13	5	5	5	5	tio	5	5	5	81	Verdana	18
14	2	3	2	3	margarida	4	4	3	23	Castellar	20
15	5	5	5	5	Alice e Margarida	5	5	5	45	Arial Narrow	24
Média	4,4	4,5	4,3	4,3	100%	4,1	4	4,1	35,1		27,6

Tabela 4.4: Resultados da Experiência 4 (Segmentação manual)

Comentário 4, com base nos resultados da tabela 4.4 verificou-se que os utilizadores, com a segmentação manual, variaram a fonte e o tamanho das letras, o tamanho médio das letras foi de 27,6. o tempo médio de leitura nesta experiência foi de 35.1 segundos, a maioria dos utilizadores consideraram que o texto era fácil de ler(4,5), fácil de seguir(4,3), gostaram de ler o texto(4,4) e perceberam o sentido da história(4,3).

4.2.1 Comparação dos resultados das experiências

Comparando as várias experiências analisadas anteriormente e tendo em conta o gráfico da figura 4.1 e apesar do conjunto limitado de experiências pode-se dizer que para uma experiência de leitura e inteligibilidade, a segmentação manual permite uma leitura mais rápida. No entanto, este resultado não permite tirar grandes conclusões visto que a segmentação manual está relacionada com a experiência 4 que é a última experiência que o utilizador faz, ou seja, na altura em que o utilizador já está mais familiarizado com a aplicação. Entretanto, a segmentação 3-3 é a que mostra uma leitura mais lenta, Tendo em conta que esta leitura permite uma maior concentração dos utilizadores e que estes optaram por ler o texto na velocidade mais lenta e isso faz com que aumente o tempo de leitura. Em cada experiência os textos são diferentes, o facto do texto ser mais fácil ou mais difícil pode influenciar a duração da leitura.

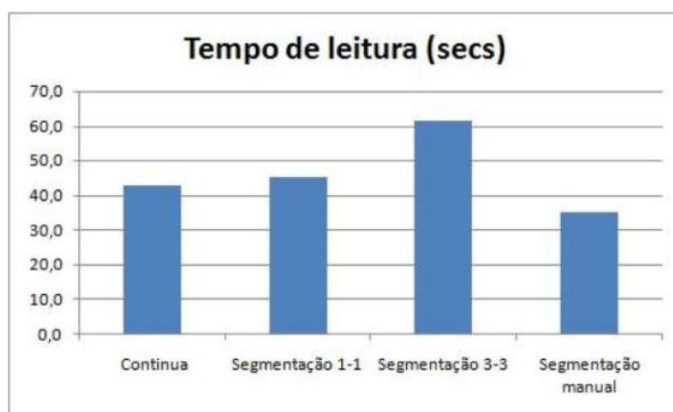


Figura 4.1: Tempo de leitura vs. Tipo de segmentação

O gráfico da figura 4.2 mostra o feedback dos utilizadores relativamente a segmentação. Com estes resultados pode-se indicar que a segmentação manual e segmentação 1-1 não degradam a experiência do utilizador, ao contrário da segmentação 3-3 que apresenta valores mais baixos.

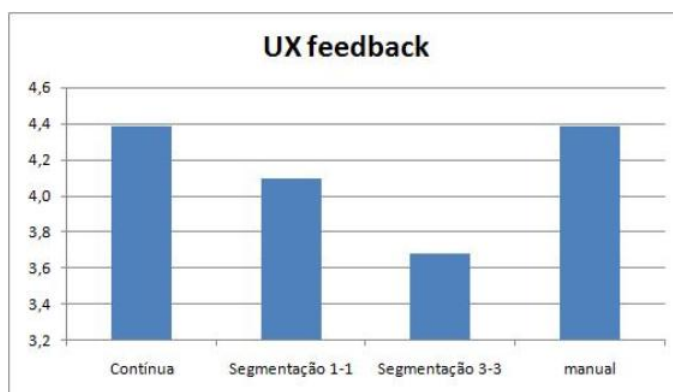


Figura 4.2: Feedback dos utilizadores relativamente a segmentação

Com os resultados das experiências há indicação de que o tipo de segmentação também influencia a opinião do utilizador em relação a qualidade da aplicação (Q6-Q8). Como se pode ver no gráfico da figura 4.3, nota-se que no caso da segmentação 1-1 os utilizadores consideram que a aplicação era fácil de se entender e que era fácil de controlar e usar, enquanto que na segmentação 3-3 o efeito foi o contrário.

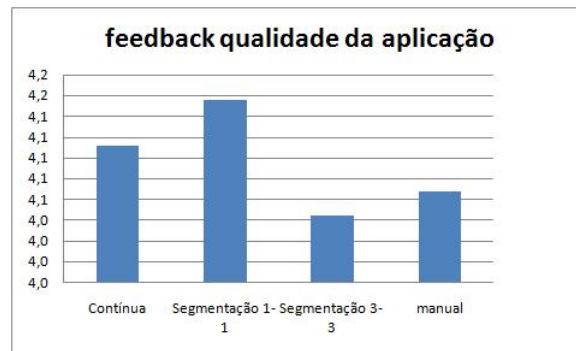


Figura 4.3: Feedback dos utilizadores relativamente a qualidade da aplicação

Com os resultados das tabelas e dos gráficos mostrados anteriormente, pode-se dizer que a segmentação manual é a que permite uma leitura mais rápida, e a segmentação 3-3 é que permite uma leitura mais lenta. A segmentação 3-3 é a que apresenta valores inferiores no que se refere a experiência dos utilizadores e é a que apresenta valores inferiores na qualidade da aplicação. Outros fatores que se deveria ter em conta na avaliação podem ser, a variação do tamanho e do tipo das letras, a variação das cores das letras e do fundo.

Capítulo 5

Conclusão

Neste documento foi apresentado o projeto desenvolvido, que consistia em criar uma ferramenta em que o seu principal objetivo é compreender e avaliar a experiência do utilizador em relação a leitura.

Começou-se por apresentar alguns conceitos de áreas relacionadas com o trabalho desenvolvido, como a UX(user experience), utilização de dispositivos de eye-tracking no apoio a leitura, alguns exemplos de aplicações relacionadas com o trabalho desenvolvido e uma análise global da leitura. Para alcançar os objetivos propostos foi criado a ferramenta TextReaderA, utilizando a linguagem Java com as funcionalidades que procura analisar os modos de leitura e avaliar a experiência de utilizador. Para analisar sua a experiência em relação a leitura, os utilizadores tinham que ler quatro textos diferentes em quatro modos de leitura diferentes, que foram, leitura contínua, com segmentação 1-1, com segmentação 3-3 e segmentação manual.

Os resultados destas experiências foram analisados através de cálculos estatísticos que permitiu perceber como é que a variação de apresentação do texto afeta a experiência do utilizador. Estes resultados mostraram que de acordo com o modo de apresentação de texto as perceções dos utilizadores em relação a aplicação variaram. Por exemplo, na segmentação manual os utilizadores acharam que os textos eram fáceis de ler e de acompanhar, enquanto que na segmentação 3-3 a maioria não gostou de ler o texto e achou que era difícil de ler e de perceber nestas condições. Também podemos observar que na segmentação 3-3 verificou uma leitura mais lenta, ou seja, um aumento do tempo de leitura. Existem outras possibilidades de exploração se ferramenta TextReaderA que não foram analisadas, uma delas é perceber se a variação das cores do texto e do fundo afeta a experiência de leitura utilizadores , por exemplo, verificar o contraste entre as cores do texto e do fundo influencia a opinião dos utilizadores em relação a sua experiência leitura. A integração da ferramenta TextReaderA em dispositivos móveis seria uma possibilidade a ser explorada na avaliação da experiência de leitura dos utilizadores. Isto pode ser uma ideia interessante para perceber como é que a utilização da ferramenta em dispositivos móveis pode afetar a experiência de leitura.

Bibliografia

- [1] Effie Lai-Chong Law, Virpi Roto, Marc Hassenzahl, Arnold P.O.S. Vermeeren, and Joke Kort. Understanding, scoping and defining user experience. page 719. ACM Press, 2009.
- [2] B.N. Schilit, M.N. Price, G. Golovchinsky, K. Tanaka, and C.C. Marshall. As we may read - the reading appliance revolution. *Computer*, 32(1):65–73, January 1999.
- [3] Stacy Lynch, Mary Nesbitt, and Steven Duke. Inside satisfaction: what it means, how to increase it. *Readeship Institute, MMC, Northwestern University*, April 2002.
- [4] Kenton O’Hara. Towards a typology of reading goals, 1996.
- [5] E. Bizas, P.G. Simos, C.J. Stam, D. Terzakis, and S. Micheloyannis. Eeg correlates of cerebral engagement in reading tasks. *Brain*, 12(2):95–105, 1999.
- [6] Stanislas Dehaene. *Reading in the brain the science and evolution of a human invention*. Viking, New York, 2009.
- [7] Julien Jung, Nelly Mainy, Philippe Kahane, Lorella Minotti, Dominique Hoffmann, Olivier Bertrand, and Jean-Philippe Lachaux. The neural bases of attentive reading. *Human Brain Mapping*, 29(10):1193–1206, October 2008.
- [8] Effie Lai-Chong Law, Marc Hassenzahl, and Ebba Thora Hvannberg. Proceedings of the workshop on towards a unified view of ux. Oslo, Norway, 2006.
- [9] User experience: Our definition. <http://www.nngroup.com/about/userexperience.html>. (último acesso em Julho 2012).
- [10] Marc Hassenzahl and Noam Tractinsky. User experience - a research agenda. *Behaviour & Information Technology*, 25(2):91–97, March 2006.
- [11] Pieter Desmet and Paul Hekkert. Framework of product experience. *International Journal of Design*, 1(1):57–66, 2007.
- [12] David Sward and Gavin Macarthur. Making user experience a business strategy. *Proceedings of the Workshop on Towards a UX Manifesto, Lencaster, 2007*, pages 30–40, 2007.

- [13] User experience network. <http://www.uxnet.org/>. (último acesso Julho de 2012).
- [14] Inês Mares, Pedro Custódio, Nuno Guimarães, Isabel Martins, Sónia Frota, and Cátia Severino. Eye-movements as markers for a cognitive processing in contínuos reading. *2nd International Conference on Eye-tracking, Visual Cognitive and Emotion*, 2(1), November 2011.
- [15] Keith Rayner. Eye movements in reading and information processing: 20 years of research. *Psychological Bulletin*, 124(3):372–422, 1998.
- [16] Alexander Pollatsek, Erik D Reichle, and Keith Rayner. Tests of the e-z reader model: exploring the interface between cognition and eye-movement control. *Cognitive psychology*, 52(1):1–56, February 2006.
- [17] Erik D. Reichle, Alexander Pollatsek, Donald L. Fisher, and Keith Rayner. Toward a model of eye movement control in reading. 1998.
- [18] Erik D. Reichle, Keith Rayner, and Alexander Pollatsek. The e-z reader model of eye-movement control in reading: Comparisons to other models. *Behavioral and Brain Sciences*, 26(04), March 2004.
- [19] Taiichi Ishida and Mitsuo Ikeda. Temporal properties of information extraction in reading studied by a text-mask replacement technique. *Journal of the Optical Society of America A*, 6(10):1624, October 1989.
- [20] App store - TEXTPlayer (AutoScroll). <http://itunes.apple.com/br/app/textplayer-autoscroll/id362291478?mt=8>. (último acesso em Agosto 2012).
- [21] App store - reader lite : Powerful eBook reader for iPhone and iPod touch. <http://itunes.apple.com/us/app/reader-lite-powerful-ebook/id346753851?mt=8>. (último acesso em Setembro 2012).
- [22] App store - reading player lite - document speed reader. <http://itunes.apple.com/us/app/reading-player-lite-document/id464750305?mt=8>. (último acesso em Setembro 2012).
- [23] Roger S Pressman. *Software engineering : a practitioner's approach*. McGraw-Hill, Boston, Mass., 2005.
- [24] Sensomotoric instruments (SMI). <http://www.smivision.com/>. (último acesso em Setembro 2012).

- [25] MATLAB- nthe language of technical computing. <http://www.mathworks.com/products/matlab/>. (último acesso em setembro 2012).
- [26] javax.swing (Java 2 platform SE v1.4.2). <http://docs.oracle.com/javase/1.4.2/docs/api/javax/swing/package-summary.html>. (último acesso Março 2012).
- [27] Eclipse - the eclipse foundation open source community website. <http://www.eclipse.org/>. (último acesso em Junho 2012).
- [28] David Gallardo, Ed Burnette, and Robert McGovern. *Eclipse in action : a guide for Java developers*. Manning, Greenwich, Conn., 2003.
- [29] XML tutorial. <http://www.w3schools.com/xml/default.asp>. (último acesso Março 2012).
- [30] org.w3c.dom (Java 2 platform SE v1.4.2). <http://docs.oracle.com/javase/1.4.2/docs/api/org/w3c/dom/package-summary.html>. (último acesso em Maio 2012).
- [31] Swing JFC java. <http://www.java2s.com/Code/Java/Swing-JFC/CatalogSwing-JFC.htm>. (último acesso Março 2012).
- [32] Lesson: Using swing components (The java™ tutorials > creating a GUI with JFC/Swing). <http://docs.oracle.com/javase/tutorial/uiswing/components/index.html>. (último acesso Março 2012).
- [33] Escala likert – wikipédia, a enciclopédia livre. http://pt.wikipedia.org/wiki/Escala_Likert. (último acesso Setembro 2012).

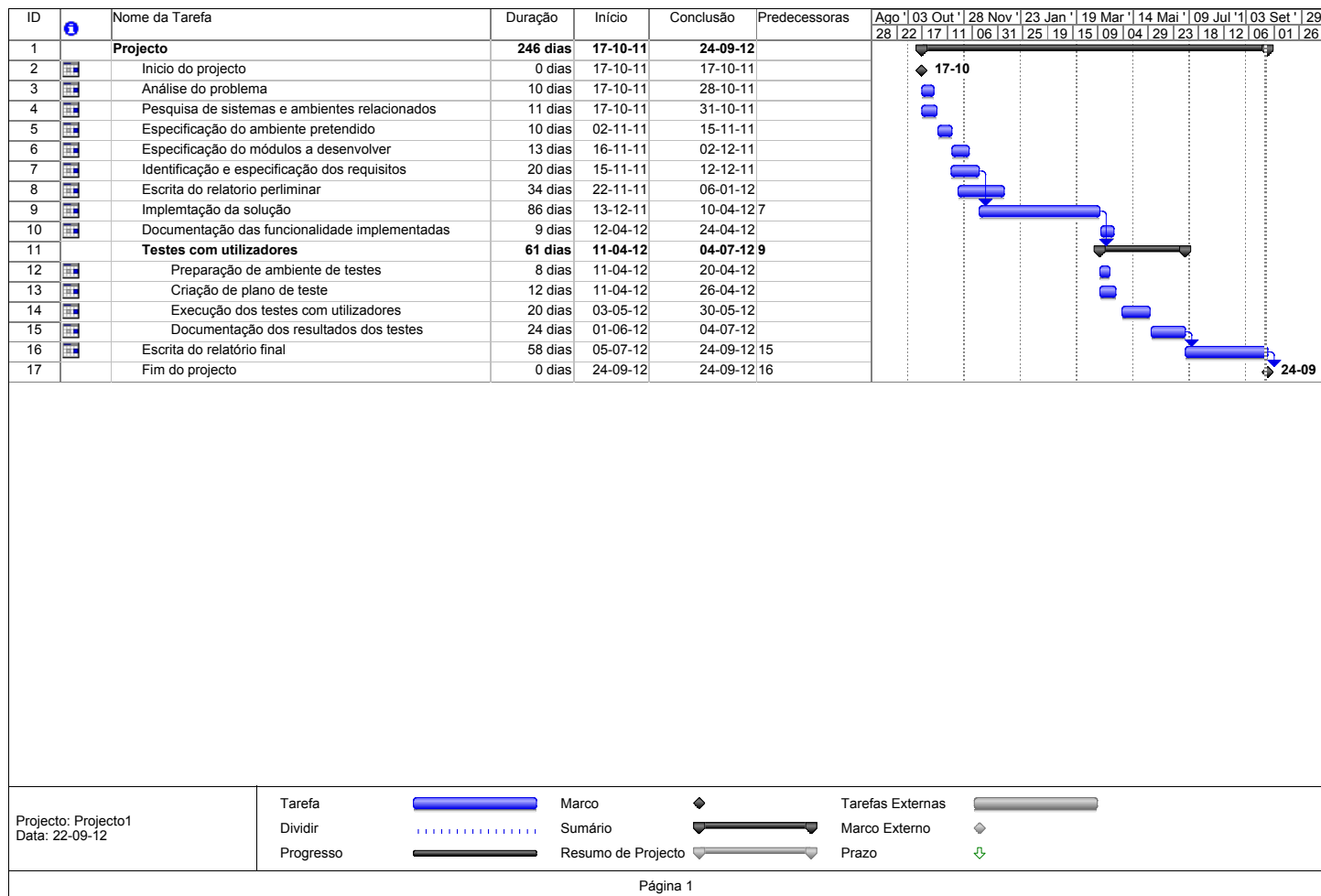
Abreviaturas

API	Application programming interface, 25
HCI	Human Computer Interface, 6
MATLAB	MATrix LABoratory, 19
RAPS	Reading Analysis with Neurophysiological Signals, 1
SMI	Sensomotoric Instruments, 19
UCD	User Centred Design, 6
UX	User eXperience, 5
WPM	Words per minute, 11

Apêndice A

Mapa de Gantt

Segue-se o mapa de Gantt com o planeamento do projecto.



Apêndice B

Poster: Análise de Modalidades e Modos de Leitura Contínua

Conferência da Associação Portuguesa de Sistemas de Informação, Guimarães 2012.

Análise de Modalidades e Modos de Leitura Contínua

Badjinca D. Batcam¹, Artemisa Moreno¹ e Nuno M. Guimarães¹

Resumo

O estudo dos processos de leitura em suporte digital tem adquirido renovada importância na sequência da disseminação de novos tipos de dispositivos e no quadro de novas preocupações como a satisfação de requisitos de acessibilidade e conceção (design) universal.

Este poster descreve duas ferramentas, ou versões de ferramentas, destinadas a facilitar a análise de aspetos particulares da leitura digital através da conceção de experiências em que são modificadas aspetos como as modalidades de suporte à leitura ou os modos de apresentação dos textos.

Introdução

A interação com sistemas computacionais e ambientes digitais em geral inclui, de forma quase universal, processos de leitura, sejam eles de reconhecimento de palavras e frases, sejam eles leitura atenta e contínua. A conceção da interface dos sistemas computacionais é determinante, por um lado para satisfazer condições de usabilidade e acessibilidade e por outro para enriquecer a experiência dos utilizadores (*user experience*, a.k.a UX).

As ferramentas apresentadas destinam-se a estudar elementos de conceção de interfaces que consideramos relevantes para a compreensão e suporte dos processos de leitura atenta e contínua. A primeira ferramenta, BookPlayerB, é na realidade um ambiente completo de reprodução de livros falados (Digital Talking Books – DTB) que, no caso presente, foi instrumentado para registo de eventos e apresentação de questionários aos utilizadores. A segunda ferramenta, TextReaderA, apresenta textos simples de acordo com diferentes estilos de segmentação.

As ferramentas permitem realizar testes de leitura atenta e contínua que discriminam variáveis de conceção como a configuração visual da interface, o uso de modalidades como o áudio, as possibilidades de segmentação dos textos. As condições experimentais foram construídas de forma a analisar o conjunto de variáveis em condições de acessibilidade limitada, nomeadamente através da utilização de um mecanismo normalizado de redução da capacidade visual.

Trabalho relacionado

A leitura e os processos cognitivos subjacentes têm sido objeto de análise e sistematização [Lynch 2002], [O'Hara 1996], embora esse quadro de referência tenha ainda pouca capacidade de operacionalização e projeção na conceção de elementos da interação.

Este trabalho reutiliza resultados de trabalhos anteriores, nomeadamente os que se realizaram no domínio dos livros falados [Serralheiro 2003], [Duarte 2007], onde a combinação estruturada de modalidades (visual e auditiva) é o fator determinante na interação.

O trabalho aqui apresentado tem como contexto mais lato dois projetos de investigação, ARIA e RAPS. O projeto ARIA – Ambient-assisted Reading Interfaces for the Ageing society, estuda e conceção de ambientes de leitura assistida para utilizadores idosos, com limitações visuais e auditivas tendencialmente mais fortes, e onde a combinação das modalidades é um aspeto central. O projeto RAPS – Reading Analysis with Physiological Signals estuda processos neurofisiológicos subjacentes à leitura atenta e contínua, sendo relevante a continuidade ou distinção entre a leitura de palavras isoladas, segmentos de texto ou texto contínuo.

Ferramentas e Ambientes de Leitura

A ferramenta BookPlayerB (fig.1) é um ambiente completo de reprodução de livros falados, instrumentado para registo de eventos e apresentação de questionários aos utilizadores.

Permite a leitura (visual) e/ou auditiva de livros (previamente lidos, gravados e sincronizados/alinhados) com elementos de animação visual e opções de adaptação visual e/ou auditiva (p.e. velocidade de reprodução).

A ferramenta TextReaderA, apresenta textos simples de acordo com diferentes estilos de segmentação. Visualmente, trata-se de uma ferramenta muito simples, constituída por um painel, no qual se apresenta texto com diferentes formas de segmentação, a saber: (I) leitura contínua; (II) segmentação 1-1, palavra a palavra; (III) segmentação 3-3, palavra a palavra com apresentação das palavras adjacentes – inspirado pela leitura parafoveal (a apresentação de palavras seguintes no campo visual periférico pode ter impacto na leitura geral) e (IV) segmentação manual ou "natural", conjuntos de palavras determinados por critérios externos que permite segmentar texto de acordo com critérios externos, em particular obtidos a partir de dispositivos eye-tracking.

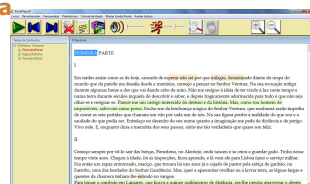


Figura 1 - BookPlayerB (janela principal)

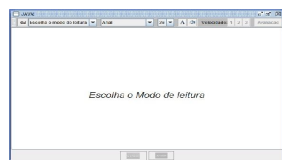


Figura 2 - TextReaderA (janela principal)

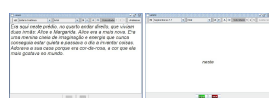


Figura 3 - TextReaderA (leitura contínua e leitura animada - com segmentação)

Experiências

As experiências realizadas têm como objetivo apurar o grau de satisfação do ponto de vista da usabilidade e acessibilidade da aplicação bem como avaliar a percepção do conteúdo do livro (no caso do BookPlayerB) e avaliar a experiência dos utilizadores (*user experience*, a.k.a UX) (no caso de TextReaderA).

Em termos de modalidades e acessibilidade estas experiências exploram as possibilidades de adaptação da apresentação de um livro oferecidas pela ferramenta BookPlayerB. Pode-se variar a dimensão das fontes de caracteres, a reprodução do texto em áudio a variadas velocidades, as cores de fundo e dos caracteres, etc.

Definiram-se quatro tipos de condições experimentais:

- (I) Leitura simples, sem sincronização (indicação explícita visual e animada do progresso da leitura), e sem áudio (reprodução do texto). Esta condição pode funcionar como controlo em análises posteriores.
- (II) Leitura sem sincronização, com áudio (reprodução do texto).
- (III) Leitura com sincronização e sucessiva degradação visual, sem áudio. A degradação visual utiliza os óculos abaixo ilustrados (Figura 4 - Cambridge Simulation Glasses) que permitem degradar, de forma normalizada, a visão dos utilizadores, desde Level 1 – perda de visão ligeira, até Level 5 – perda de visão severa).
- (IV) Leitura com sincronização, sucessiva degradação visual e modalidade áudio em conjunto.



Level of Impairment	Glasses worn (assuming wearer starts with VA 20 / 20)	Approximate Snellen Visual Acuity	Description (based on Int. Council of Ophthalmology)
None	None	20 / 20	
Level 1	[Image]	20 / 25	Mild vision loss
Level 2	[Image]	20 / 40	Borderline for driving
Level 3	[Image]	20 / 60	Moderate vision loss
Level 4	[Image]	20 / 100	
Level 5	[Image]	20 / 200	Severe vision loss

Nas experiências de leitura segmentada, o cenário é o seguinte:

1. O sujeito é apresentado à ferramenta e efetua um breve teste das funcionalidades;
2. O sujeito lê sucessivamente quatro textos diferentes, usando as várias formas de segmentação referidas (leitura contínua, segmentação 1-1, segmentação 3-3, segmentação manual);
3. No fim de cada passo responde a um questionário simples (Figura 6);

Resultados

No BookPlayerB participaram em experiências 28 pessoas de diferentes níveis de escolaridade (do básico ao superior). No TextReaderA participaram 15 pessoas com escolaridade superior.

No que se refere a modalidades e acessibilidade, os resultados são os seguintes: (a) Em condições de inteligibilidade equivalentes (com o nr de sujeitos por experiência indicado), a progressão de aumento da dimensão dos caracteres ocorre como esperado; (b) a introdução de áudio induz um aumento superior na dimensão dos caracteres, resultado aparentemente contraditório que deve ser confirmado com maior número de sujeitos.

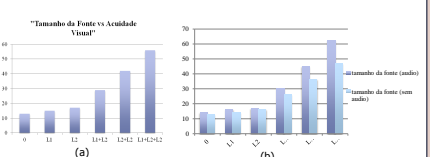


Figura 5 – (a) aumento de dimensão dos caracteres vs acuidade visual; (b) aumento da dimensão dos caracteres vs acuidade visual (com áudio)

No domínio da segmentação, a leitura contínua, a segmentação 1-1 e manual não degradam a experiência do utilizador, ao contrário da segmentação 3-3 que apresenta valores inferiores. Para UX e inteligibilidade equivalentes, a segmentação manual permite uma leitura mais rápida.

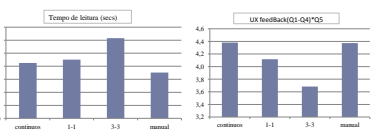


Figura 6 – Feedback dos utilizadores relativamente à segmentação; (b) tempo de leitura vs. Tipo de segmentação

Conclusão

As ferramentas (BookPlayerB e TextReaderA) desenvolvidas, adaptadas e instrumentadas com questionários e registo de dados de utilização e configuração permitem tirar conclusões específicas sobre o uso das modalidades em condições de diferente acessibilidade e sobre o impacto dos modos de segmentação na inteligibilidade da leitura. Os resultados obtidos sugerem a extensão dos painéis de sujeitos experimentais de modo a reforçar a análise estatística e permitir a formulação de questões complementares relativas a modalidades e modos de leitura.

References

- Lynch, S., Nesbitt, M. and Duke, S. (2002), Inside Satisfaction: what it means, how to increase it, Readership Institute, MMC, Northwestern University, USA, 2002
- O'Hara, K. (1996) Towards a Typology of Reading Goals, TR EPC-1996-107, Rank Xerox Research Centre, Cambridge, UK, 1996
- Serralheiro, A., Trancoso, I., Caseiro, D., Chambel, T., Carrigo, L., Guimarães, N. (2003) Towards a Repository of Digital Talking Books, Eurospeech, 2003, Geneva, Switzerland
- Duarte, C., Carrigo, L. and Guimarães, N. (2007) Evaluating usability improvements by combining visual and audio modalities in the interface. Proceedings of HCI 2007, 12th International Conference on Human Computer Interaction, LNCS Digital Library (LNCS, <http://www.springer.com/lncs>), July 2007, Beijing, China, ISBN: 978-3-540-73104-7

Apêndice C

Textos utilizados nas experiências

Texto1

Era aqui neste prédio, no quarto andar direito, que viviam duas irmãs: Alice e Margarida. Alice era a mais nova. Era uma menina cheia de imaginação e energia que nunca conseguia estar quieta e passava o dia a inventar coisas. Adorava a sua casa porque era cor-de-rosa, a cor que ela mais gostava no mundo.

Texto2

Já estava escuro quando o pintor foi para casa. No seu caminho viu uma senhora. Como lhe pareceu perdida quis saber para onde queria ir. A senhora disse que se mudara para ali havia pouco tempo mas que ainda não sabia o caminho para a sua futura casa. O pintor decidiu ajudar e levá-la a casa. Ao longo do caminho falaram um pouco e ele contou que era seu vizinho.

Texto3

Assim que o tio chegou à sala a família começou a unir-se. A mãe chegou com os pratos e pôs a mesa. O filho trouxe a sopa mas a sopa estava muito quente para comer. Durante o jantar o tio disse porque veio. Contou que fez uma venda única com grandes lucros. A família quis logo saber do que se tratava e o que vendeu. O tio não quis contar.

Texto4

Era de manhã quando o policia apagou o quadro. Foi à escola para contar aos alunos o que fazia. Apesar da sua calma quando viu os alunos não sabia o que dizer. Ficou sem palavras com a atenção dos alunos. No fim das aulas o policia foi brincar com eles e contou algumas coisas que fazia. No final da tarde já estava mais calmo para falar.

Apêndice D

Código fonte

- **MainClass.java**

```
1  package UI;
2
3  /**
4   *
5   * @author Artemisa Moreno
6   *
7   */
8  public class MainClass {
9
10     public static void main(String args[]) {
11         javax.swing.SwingUtilities.invokeLater(new Runnable
12             () {
13             public void run() {
14                 PainelPrincipal.createAndShowGUI();
15             }
16         });
17     }
```

Listagem D.1: MainClass

- **PainelPrincipal**

```
1      /**
2       * Inicializa os componentes da aplicacao
3       */
4      public void inicializar() {
5          /**
6           * inicializa as opcoes para aresentar de textos
7           */
8          currentDirectory = new File(".");
9          context = new StyleContext();
10         document = new DefaultStyledDocument(context);
11         style = context.getStyle(StyleContext.DEFAULT_STYLE)
12             ;
```

```
12         StyleConstants.setAlignment(style, StyleConstants.
13             ALIGN_CENTER);
14         StyleConstants.setSpaceAbove(style, getHeight() / 2
15             - 50);
16         this.addComponentListener(new ComponentAdapter() {
17             public void componentResized(java.awt.event.
18                 ComponentEvent e) {
19                 StyleConstants.setSpaceAbove(style,
20                     getHeight() / 2 - 50);
21             }
22         });
23         // criar file chooser
24         fc = new JFileChooser();
25         // painel onde vao ser apresentados os botoes
26         painelBotoes = new JPanel();
27         // criar toolbar onde vai ser inserido o menu
28         toolBar = new JToolBar();
29         toolBar.setBackground(Color.LIGHT_GRAY);
30         criarBotoes();
31         criarComboBoxes();
32
33         toolBar.add(buttonCorTexto);
34         toolBar.add(buttonCorFundo);
35         toolBar.addSeparator();
36         toolBar.add(new JLabel("Velocidade:"));
37         toolBar.add(vel1);
38         toolBar.add(vel2);
39         toolBar.add(vel3);
40         toolBar.addSeparator();
41         toolBar.add(aval);
42         painelBotoes.add(start, BorderLayout.CENTER);
43         painelBotoes.add(stop, BorderLayout.CENTER);
44
45         try {
46             document.insertString(document.getLength(),
47                 textoInicial, style);
48         } catch (BadLocationException e) {
49             e.printStackTrace();
50         }
51         textPane = new JTextPane(document);
52         textPane.setEditable(false);
53         textPane.setFont(new Font((String) comboFont.
54             getSelectedItem(),
55                 Font.ITALIC, size));
56
57         JScrollPane painelTexto = new JScrollPane(textPane);
58         // adicionar os componentes
59         getContentPane().add(toolBar, BorderLayout.NORTH);
```

```

55         getContentPane().add(painelBotoes, BorderLayout.
           SOUTH);
56         getContentPane().add(painelTexto, BorderLayout.
           CENTER);
57     }

```

Listagem D.2: Código 2: PainelPrincipal.inicializar()

```

1
2     /**
3      * Atualiza o painel de texto de acordo com o tipo de
4      * apresentacao do texto passado com parametro
5      * @param seg, o tipo de apresentação
6      */
7     public void setPainelTexto(String seg) {
8
9         if (seg.equals("Leitura Continua")) {
10             vel1.setEnabled(false);
11             vel2.setEnabled(false);
12             vel3.setEnabled(false);
13             start.setEnabled(false);
14             stop.setEnabled(false);
15             aval.setEnabled(true);
16             lerTxt(file);
17             stopTimer();
18             StyleConstants.setAlignment(style,
19                 StyleConstants.ALIGN_LEFT);
20             textPane.setDocument(new ModoContinuo(texto).
21                 getDocument());
22             time = System.currentTimeMillis();
23
24         } else if (!seg.equals(TIPOSEGMENTACAO[0])) {
25             start.setEnabled(true);
26             vel1.setEnabled(true);
27             vel2.setEnabled(true);
28             vel3.setEnabled(true);
29             aval.setEnabled(false);
30             StyleConstants.setAlignment(style,
31                 StyleConstants.ALIGN_CENTER);
32
33             try {
34                 document = new DefaultStyledDocument(context
35                     );
36                 document.insertString(document.getLength(),
37                     textoSegs, style);
38                 textPane.setDocument(document);
39
40             } catch (BadLocationException e1) {
41                 e1.printStackTrace();
42             }
43         }
44     }

```

```

37         if (seg.equals("Segmentacao 1-1")) {
38             lerTxt(file);
39             stopTimer();
40             document = new DefaultStyledDocument(context
41                 );
42             Segmentacao1 segment = new Segmentacao1(
43                 document, texto);
44             timer = new Timer(delay, segment.tratarTexto
45                 ());
46         } else if (seg.equals("Segmentacao 3-3")) {
47             lerTxt(file);
48             stopTimer();
49             document = new DefaultStyledDocument(context
50                 );
51             Segmentacao3 segment = new Segmentacao3(
52                 document, texto);
53             timer = new Timer(delay, segment.tratarTexto
54                 ());
55         } else if (seg.equals("Segmentacao Manual")) {
56             stopTimer();
57             document = new DefaultStyledDocument(context
58                 );
59             SegManual manual = new SegManual(file,
60                 document);
61             timer = new Timer(delay, manual.tratarTexto
62                 ());
63         }
64     }
65 }

```

Listagem D.3: PainelPrincipal.setPainelTexto(String seg)

```

1  /**
2   * Cria e visualiza os componentes da aplicacao
3   */
4  static void createAndShowGUI() {
5      // Make sure we have nice window decorations.
6      JFrame.setDefaultLookAndFeelDecorated(true);
7
8      // Create and set up the window.
9      PainelPrincipal frame = new PainelPrincipal();
10     frame.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE)
11         ;
12
13     frame.setSize(700, 500);
14     // frame.setResizable(false);
15     Dimension screenSize = Toolkit.getDefaultToolkit().
16         getScreenSize();
17     frame.setLocation(screenSize.width / 4, screenSize.

```

```
        height / 4);  
16  
17        frame.inicializar();  
18        frame.setFocusable(true);  
19        frame.setVisible(true);  
20    }
```

Listagem D.4: PannelPrincipal.createAndShowGUI()